

UNISYS

**B 2000/B 3000/
B 4000/V Series**

DMSII

**Operations
Reference Manual**

Volume 3: Utilities

Copyright © 1987 Unisys Corporation
All Rights Reserved

Unisys is a trademark of Unisys Corporation

Relative to Release
Level 7.2

Priced Item

December 1987
Distribution Code SD
Printed in U S America
5026586

TABLE OF CONTENTS

Section	Title	Page
	ABOUT THIS MANUAL	xiii
	PURPOSE, SCOPE, AND AUDIENCE	xiii
	ORGANIZATION	xiii
	RELATED DOCUMENTS	xiv
1	GENERAL UTILITY INFORMATION	1-1
	INTRODUCTION	1-1
	REQUIRED FILES	1-2
	USING DMSUTL WITH CANDE	1-3
	USING DMSUTL WITH GEMCOS	1-4
	USING DMSUTL WITH MC7	1-5
	USING DMSUTL WITH SWITCH	1-6
	USING THE DMSUTL HELP SCREENS	1-10
	General Help Screen Procedure	1-10
	DMSUTL Help Screen Procedure	1-10
	Freeform Syntax Help Screen Procedure	1-10
	USING THE DMSUTL DATA ENTRY SCREENS	1-11
	Common Screen Choices	1-11
	Making Selections and Checking Fields	1-12
	HOME MENU	1-12
	USING DMSUTL IN BATCH MODE	1-14
	Executing DMSUTL from the ODT	1-14
	Executing DMSUTL from Cards or Card Images	1-14
	Initiating DMSUTL from WFL	1-14
	PARAMETERS	1-15
2	DATABASE PROGRAM DUMP ANALYSIS UTILITY (DBPDMP)	2-1
	INTRODUCTION	2-1
	INITIATING DBPDMP - SCREEN MODE	2-1
	FREEFORM SYNTAX	2-4
	INITIATING DBPDMP - STAND-ALONE MODE	2-5
	With the PM Command	2-5
	From the ODT	2-5
	From Cards or Card Images	2-5
	From a WFL Job	2-5
	EXAMPLES	2-6
3	STATISTICS REPORTING UTILITY (DMSTAT)	3-1
	INTRODUCTION	3-1
	INITIATING DMSTAT - SCREEN MODE	3-1
	FREEFORM SYNTAX	3-4
	INITIATING DMSTAT - BATCH MODE	3-4
	DMSTAT REPORT OPTIONS	3-6
	Database Report	3-6
	Structure Report	3-8
	User Report	3-16
4	STRUCTURE LISTING UTILITY (DMSLST)	4-1
	INTRODUCTION	4-1

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	LISTING DATABASE FILES - SCREEN MODE	4-1
	FREEFORM SYNTAX	4-4
	LISTING DATABASE FILES - BATCH MODE	4-4
	LISTING ISAM FILES - SCREEN MODE	4-6
	FREEFORM SYNTAX	4-8
	LISTING ISAM FILES - BATCH MODE	4-8
	SAMPLE REPORTS	4-10
5	CONTROL FILE PRINT UTILITY (DMSCTL)	5-1
	INTRODUCTION	5-1
	INITIATING DMSCTL - SCREEN MODE	5-1
	FREEFORM SYNTAX	5-2
	INITIATING DMSCTL - BATCH MODE	5-3
	DMSCTL REPORT	5-3
6	DATABASE ANALYSIS UTILITY (DMSDAP)	6-1
	INTRODUCTION	6-1
	INITIATING DMSDAP - SCREEN MODE	6-1
	FREEFORM SYNTAX	6-3
	INITIATING DMSDAP - BATCH MODE	6-3
	DESCRIPTION OF DMSDAP ANALYSIS	6-4
	DMSDAP REPORT	6-5
7	DDF PRINT UTILITY (DDFLIS)	7-1
	INTRODUCTION	7-1
	INITIATING DDFLIS - SCREEN MODE	7-1
	FREEFORM SYNTAX	7-2
	INITIATING DDFLIS - BATCH MODE	7-3
8	AUDIT TRAIL UTILITY (DMSAUD)	8-1
	INTRODUCTION	8-1
	INITIATING DMSAUD - SCREEN MODE	8-1
	FREEFORM SYNTAX	8-6
	INITIATING DMSAUD - STAND-ALONE MODE	8-7
	From the ODT	8-7
	From Cards or Card Images	8-7
	From a WFL Job	8-7
	PARAMETERS	8-8
	DMSAUD REPORTS	8-11
	DESCRIPTION OF CONSISTENCY ANALYSIS	8-17
9	SOURCE GENERATION UTILITY (DMSGEN)	9-1
	INTRODUCTION	9-1
	INITIATING DMSGEN - SCREEN MODE	9-1
	FREEFORM SYNTAX	9-3
	INITIATING DMSGEN - BATCH MODE	9-4
	UNLOAD FUNCTION	9-5
	Generating an UNLOAD Program	9-5
	Format of Flat Files	9-6
	LOAD FUNCTION	9-6

TABLE OF CONTENTS (Cont.)

Section	Title	Page
	Generating a LOAD Program	9-6
	SKELETON FUNCTION	9-7
	Generating a SKELETON Program	9-7
10	DATABASE MAINTENANCE UTILITY (DMSDLR)	10-1
	INTRODUCTION	10-1
	INITIATING DMSDLR FUNCTIONS - SCREEN MODE	10-2
	HALT/LOAD FUNCTION	10-3
	REBUILD FUNCTION	10-3
	ROLLBACK FUNCTION	10-4
	RECONSTRUCT FUNCTION	10-5
	REORGANIZE FUNCTION	10-5
	PRODUCTION FUNCTION	10-7
	PURGE FUNCTION	10-7
	LOAD FUNCTION	10-7
	DUMP FUNCTION	10-8
	RESTART FUNCTION	10-8
	USING DMSDLR FUNCTIONS - FREEFORM SYNTAX	10-8
	INITIATING DMSDLR - STAND-ALONE MODE	10-13
11	UTILITY MESSAGES	11-1
	INDEX	1

LIST OF ILLUSTRATIONS

Figure	Title	Page
1-1	DMSUTL Home Menu Screen	1-12
1-2	Parameters for Using DMSUTL in Batch Mode	1-15
1-3	Database Specifications for Using DMSUTL in Batch Mode	1-16
2-1	DBPDMP Menu Screen	2-2
2-2	DMSUTL Freeform Syntax for DBPDMP	2-4
3-1	DMSTAT Menu Screen	3-2
3-2	Freeform Syntax for DMSTAT	3-4
3-3	DMSTAT Specification for Batch Use	3-4
3-4	DMSTAT Parameters for Batch Use	3-5
3-5	Sample of DMSTAT Database Report	3-6
3-6	Sample of DMSTAT Structure Report	3-9
3-7	Sample of DMSTAT User Report	3-17
4-1	DMSLST Menu Screen	4-2
4-2	DMSUTL Freeform Syntax for Listing Database Files with DMSLST ..	4-4
4-3	DMSLST Specification for Batch Use	4-4
4-4	DMSLST Parameters for Listing Database Files - Batch Mode	4-5
4-5	DMSLST ISAM Menu Screen	4-6
4-6	DMSUTL Freeform Syntax for Listing ISAM Files with DMSLST	4-8
4-7	DMSLST Parameters for Listing ISAM Files from ODT or Card Input ..	4-8
4-8	Sample Copy of DMSLST Control Report	4-10
4-9	Sample Copy of DMSLST Block Report	4-11
4-10	Sample Copy of DMSLST Record Report	4-12
4-11	Sample Copy of DMSLST BCI Report	4-13
4-12	Sample Copy of DMSLST Block One Report	4-14
4-13	Sample Copy of DMSLST Summary Report	4-15
5-1	DMSCTL Menu Screen	5-1
5-2	DMSUTL Freeform Syntax for DMSCTL	5-2
5-3	DMSCTL Specification for Batch Use	5-3
5-4	Sample of DMSCTL Report	5-4
6-1	DMSDAP Menu Screen	6-2
6-2	DMSUTL Freeform Syntax for DMSDAP	6-3
6-3	DMSDAP Specification for Batch Use	6-3
7-1	DDFLIS Menu Screen	7-1
7-2	DMSUTL Freeform Syntax for DDFLIS	7-2
7-3	DDFLIS Specification for Batch Use	7-3
8-1	DMSAUD Menu Screen	8-2
8-2	DMSAUD Record Selection Screen	8-4
8-3	DMSUTL Freeform Syntax for DMSAUD	8-6
8-4	DMSAUD Parameters for Batch Use	8-8
8-5	Sample Listing from DMSAUD	8-13
9-1	DMSGEN Menu Screen	9-2
9-2	DMSUTL Freeform Syntax for DMSGEN	9-3
9-3	DMSGEN Specification for Batch Use	9-4
9-4	DMSGEN Parameters for Batch Use	9-4
10-1	DMSDLR Menu Screen	10-2
10-2	DMSUTL Freeform Syntax for DMSDLR	10-9

LIST OF TABLES

Table	Title	Page
1-1	DMSUTL Required Files	1-2
1-2	TRANSACTION Definition	1-7
1-3	HANDLER Definition	1-8
1-4	PROCESS Definition	1-9
8-1	Audit Record Types	8-5

ABOUT THIS MANUAL

PURPOSE, SCOPE, AND AUDIENCE

The Data Management System II (DMSII) is a software product that enables a Database Administrator (DBA) to create and maintain a database. This manual describes each of the following DMSII utilities: DBPDMP, DMSTAT, DMSLST, DMSCTL, DMSDAP, DDFLIS, DMSAUD, DMSGEN, and DMSDLR. In addition, this manual contains all of the system messages and error messages that pertain to DMSUTL.

This manual is written for DMSII operators. This manual is part of a three volume set that includes the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* and the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*. To use this manual, first become completely familiar with the utilities that you will use.

ORGANIZATION

This document consists of the following 11 sections:

SECTION 1: GENERAL UTILITY INFORMATION

This section provides a general overview of the DMSII utility functions.

SECTION 2: DATABASE PROGRAM DUMP ANALYSIS UTILITY (DBPDMP)

This section describes the Database Program Dump Analysis Utility (DBPDMP).

SECTION 3: STATISTICS REPORTING UTILITY (DMSTAT)

This section describes the Statistics Reporting Utility (DMSTAT).

SECTION 4: STRUCTURE LISTING UTILITY (DMSLST)

This section describes the Structure Listing Utility (DMSLST).

SECTION 5: CONTROL FILE PRINT UTILITY (DMSCTL)

This section describes the Control File Print Utility (DMSCTL).

SECTION 6: DATABASE ANALYSIS UTILITY (DMSDAP)

This section describes the Database Analysis Utility (DMSDAP).

SECTION 7: DDF PRINT UTILITY

This section describes the DDF Print Utility (DDFLIS).

SECTION 8: AUDIT TRAIL UTILITY (DMSAUD)

This section describes the Audit Trail Utility (DMSAUD).

SECTION 9: SOURCE GENERATION UTILITY (DMSGEN)

This section describes the Source Generation Utility (DMSGEN).

SECTION 10: DATABASE MAINTENANCE UTILITY (DMSDLR)

This section describes the DMSDLR Utility (DMS Dump, Load, and Recovery) including the DMSREO utility for reorganization.

SECTION 11: UTILITY MESSAGES

This section lists each of the system messages and error messages that relate to DMSUTL.

RELATED DOCUMENTS

The following related publications are listed, for reference, according to the operating system to which they apply.

B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

B 2000/B 3000/B 4000/V Series DMSII Inquiry Operations Guide and the *DMINQUIRY Reference Card* for MCPIX and MCP/VS 1.0; or the *B 2000/B 3000/B 4000/V Series DMSII Inquiry Operations Guide* for MCP/VS 2.0.

B 2000/B 3000/B 4000 Series MCPIX System Software Operations Guide Volume 1: System Commands and *B 2000/B 3000/B 4000 Series MCPIX System Software Operations Guide Volume 2: System Utilities* for MCPIX or *V Series MCP/VS System Software Operations Guide Volume 2: System Commands* and *V Series MCP/VS System Software Operations Guide Volume 3: System Utilities* for MCP/VS 2.0.

B 2000/B 3000/B 4000 Series MCPIX System Software Operations Guide Volume 1: System Commands for MCPIX or *V Series MCP/VS System Software Operations Guide Volume 1: Installation* for MCP/VS 2.0.

B 2000/B 3000/B 4000/V Series CANDE Installation and Operation Guide for MCPIX and MCP/VS 1.0; or the *V Series CANDE Installation and Operations Guide* for MCP/VS 2.0.

B 2000/B 3000/B 4000 Series COBOL ANSI-74 Compiler Programming Reference Manual for MCPIX and MCP/VS 1.0; or the *V Series COBOL ANSI-74 Compiler Programming Reference Manual* for MCP/VS 2.0.

B 2000/B 3000/B 4000/V Series Pascal Compiler Programming Reference Manual for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

B 2000/B 3000/B 4000/V Series RPG Compiler Programming Reference Manual for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

B 2000/B 3000/B 4000/V Series Programmer's Guide for MCPIX and MCP/VS 1.0; or the *V Series MCP/VS Programming Reference Manual* for MCP/VS 2.0.

SECTION 1

GENERAL UTILITY INFORMATION

INTRODUCTION

This section describes each of the following Data Management System II (DMSII) utility functions:

- DBPDMP, which produces an analysis listing of program dump files from database programs (DBPs).
- DMSTAT, which produces statistical reports on database usage and enables you to reinitialize the database statistics file.
- DMSLST, which lists database structure files, and which also lists non-database Indexed Sequential-Access Method (ISAM) files.
- DMSCTL, which produces a report of the information in a database control file.
- DMSDAP (Database Analysis Program), which checks the integrity of a database.
- DDFLIS, which produces a report of the information in a Data Definition File (DDF).
- DMSAUD, which enables you to print, verify, or copy the information in an audit trail.
- DMSGEN, which generates COBOL74 source for three kinds of programs:
 - Programs that unload data from a database to flat disk or pack files
 - Programs that load data into a database from previously unloaded files
 - Skeleton database programs that you can expand upon. In skeleton programs, some of the more tedious coding tasks are done automatically.
- DMSDLR, which performs dumping, loading, recovery, purging and reorganization functions.

There are several ways to use these utilities:

- You can use the DMSUTL utility in screen mode from data communications terminals. This interface displays menu screens that prompt input and provide access to all of the utility functions. The following portions of this section describe this interface: "Using DMSUTL with CANDE," "Using DMSUTL with GEMCOS," "Using DMSUTL with MC7," and "Using DMSUTL with SWITCH." CANDE, GEMCOS, MC7, and SWITCH are Unisys data communications systems; select the one that you use at your facility.

- You can use the DMSUTL utility in batch mode. You enter commands at the ODT, from a card deck, or from a Work Flow Language (WFL) program. This interface provides access to the DMSTAT, DMSLST, DMSCTL, DMSDAP, DDFLIS, and DMSGEN functions to users without access to a data communications terminal, and enables you to perform regular initiation with WFL. For more information about batch use, refer to the "Using DMSUTL in Batch Mode" portion of this section.
- DMSAUD, DBPDMP, and DMSDLR are stand-alone programs that you can use independently of DMSUTL. The discussion of the IR command (see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*) gives you instructions on how to use DMSDLR in this manner. Instructions for using DMSAUD and DBPDMP in this manner appear in the parts of this manual that discuss those functions. You can use these programs in WFL programs and without a data communications terminal.

REQUIRED FILES

Table 1-1 shows the files that must reside on disk before you can use the various functions of DMSUTL. These files come with the DMSII release.

Table 1-1. DMSUTL Required Files

File Name	File Type	When Needed
DMSUTL	Code	Always needed for using the DMSUTL interface.
DMSFRM	Data	Needed if you use the DMSUTL utility from a data communications terminal.
DMSMSG	Data	
DMSAUD	Code	Needed if you use the DMSAUD utility.
DBPDMP	Code	Needed if you use the DBPDMP utility.
DMSDLR	Code	Needed if you use the DMSDLR utility.
DMSAPP	Code	Needed if you use the DMSGEN utility.
DMSSKL	Data	
DMSREO	Code	Needed if you use the REORGANIZE option of the DMSDLR utility.

USING DMSUTL WITH CANDE

CANDE is the Unisys Command and Edit system, a time-sharing data communications system. For full information on CANDE, refer to the *B 2000/B 3000/B 4000/V Series CANDE Installation and Operation Guide* or the *V Series CANDE Installation and Operations Guide* (for MCP/VS 2.0).

To use DMSUTL with CANDE, proceed as follows:

1. Make sure the files that appear in Table 1-1 reside on disk. These files come with the DMSII release.
2. Log onto CANDE at your terminal, by entering:

LI <usercode> <password>

- 3a. You can initiate the CANDE EDITOR and then initiate DMSUTL as a time-sharing process, by entering:

RUN EDITOR
SYS DMSUTL

- 3b. You can initiate DMSUTL directly with the RUN DMSUTL command, if you have previously declared the program DMSUTL in the CANDE initialization parameters. The record in the CANDE initialization file would look like this:

PROGRAM DMSUTL

Refer to the discussions of CANDE initialization in the *B 2000/B 3000/B 4000/V Series CANDE Installation and Operation Guide* or the *V Series CANDE Installation and Operations Guide* (for MCP/VS 2.0).

To initiate DMSUTL, enter the following at the CANDE terminal:

RUN DMSUTL

To sum up, the following command sequences at the CANDE terminal will initiate DMSUTL:

LI <usercode> <password>
RUN EDITOR
SYS DMSUTL

LI <usercode> <password>
RUN DMSUTL

When you enter these commands, the DMSUTL home menu screen appears on your terminal. A discussion of the home menu screen appears in the "Home Menu" portion of this section.

USING DMSUTL WITH GEMCOS

GEMCOS is the Unisys Generalized Message Control System. A full description of GEMCOS appears in the *B 2000/B 3000/B 4000/V Series GEMCOS Implementation and Programming Reference Manual*.

To use DMSUTL with GEMCOS, proceed as follows:

1. Make sure the files that appear in Table 1-1 reside on disk. These files come with the DMSII release.

With a single DMSUTL code file, only one GEMCOS user can be signed onto DMSUTL at a time. To enable multiple GEMCOS users to initiate DMSUTL at the same time, multiple copies of the DMSUTL code file must exist on disk under different names. The names DM1UTL, DM2UTL, and so on, are suggested for the multiple copies of DMSUTL. In no case should you use the names DMSU72 or DMSUXX because they would interfere with other functions. You must declare each copy in TCL (see step 2).

2. A program definition for each copy of DMSUTL must appear in the Definition section of the Transaction Control Language (TCL) input, also called the specification file. The following is a sample TCL program definition for DMSUTL.

```
PROGRAM DMSUTL (16):  
ATTRIBUTES           = "/GEMCOS/<mcs stoqnameout> PR 6 ".  
EOJ                  = FALSE.  
EXECUTE              = ON DEMAND.  
MAXCOPIES            = 1.  
INTERFACE            = STOQ.  
INTERFACESIZE        = 2100.  
MYUSE                = IO.  
FASTPATH             = IO.
```

If multiple copies of DMSUTL reside on disk, as described in step 1, you must declare each copy in TCL.

3. The terminal must be assigned to DMSUTL. You can do this with one of the following assignment commands:

*ASSIGN TO PROGRAM DMSUTL

OR

*CONNECT DMSUTL

Where:

* = the local ECC character

4. To initiate DMSUTL, transmit a blank. The DMSUTL home menu screen will appear. A discussion of the home menu screen appears in the "Home Menu" portion of this section.

USING DMSUTL WITH MC7

MC7 is a Unisys message control system generator. A description of MC7 appears in the *B 774 Message Control System Generator Reference Manual*. To use DMSUTL with MC7, proceed as follows:

1. Make sure the files that appear in Table 1-1 reside on disk. These files come with the DMSII release.

With a single DMSUTL code file, only one MC7 user can be signed onto DMSUTL at a time. To enable multiple MC7 users to initiate DMSUTL at the same time, multiple copies of the DMSUTL code file must exist on disk or diskpack under different names. The names DM1UTL, DM2UTL, and so on, are suggested for the multiple copies of DMSUTL. In no case should you use the names DMSU72 or DMSUXX because they would interfere with other functions. You must declare each copy in the MC7 generation source file (see step 2).

2. The program section of the MC7 generation source file must include each copy of DMSUTL. The following is a sample program section for DMSUTL.

PROGRAM SECTION.

IDENTITY	= "DMSUTL".
PRIORITY	= 7.
ZIPEXECUTE	= FIRST DEMAND.
ZIPSTART	= ACTIVITY.
LOCK	= TRUE.
INTERFACE	= SEND-RECEIVE.
MAXSEND	= 2000.
MAXRECEIVE	= 2000.
CHARGE	= 000001.
QUEUEING	= FAST-TANK.
STOQUE	= TRUE.

3. The device section of the MC7 generation source file must declare a buffer size of 300 bytes.
4. The terminal must be assigned to DMSUTL. For information on assignment, refer to the *B 774 Message Control System Generator Reference Manual*.
5. To run DMSUTL under MC7, you must patch the DMSUTL code file as follows:

PATCH DMSUTL X0 8 12 UA "MC7MCSSTQOUT"

Where:

MC7MCS = the name of the MC7 MCS

STQOUT = the output storage queue name

6. To initiate DMSUTL, transmit a blank. The DMSUTL home menu screen will appear. A discussion of the home menu screen appears in the "Home Menu" portion of this section.

USING DMSUTL WITH SWITCH

SWITCH is a Message Control System that runs on the CP3680 front-end processor.

To use DMSUTL with SWITCH, proceed as follows:

1. Make sure the files that appear in Table 1-1 reside on disk. These files come with the DMSII release:

With a single DMSUTL code file, only one SWITCH user can be signed onto DMSUTL at a time. To enable multiple SWITCH users to initiate DMSUTL at the same time, multiple copies of the DMSUTL code file must exist on disk under different names. The names DM1UTL, DM2UTL, and so on, are suggested for the multiple copies of DMSUTL. In no case should you use the names DMSU72 or DMSUXX because they would interfere with other functions. You must declare a TRANSACTION, HANDLER, and PROCESS for each copy.

2. To run DMSUTL under SWITCH, you must create the TRANSACTION, HANDLER, and PROCESS definitions ("x" and "n" indicate site-dependent information) that appear in Tables 1-2 through 1-4.
3. To run DMSUTL under SWITCH, you must patch DMSUTL code file as follows:

```
PATCH DMSUTL X0 8 12 UA "SWITCHNDSTOQ"
```

Where

SWITCH = the name of the CP3680 interface program

NDSTOQ = the output storage queue name

The reason for this is that DMSUTL's output STOQ name is supplied through the second slash parameter.

4. To initiate DMSUTL under SWITCH, enter:

```
/DMSUTL
```

The DMSUTL home menu will appear on your screen.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
General Utility Information

Table 1-2. TRANSACTION Definition

USER: xxxxxxxxxxxx

CP3680 RESIDENT HANDLER
VIEW TRANSACTION nnnn

DATE: MM/DD/YY
FORM: xxxx

TRANSACTION NAME	DMSUTL
SECURITY GROUP NAME	xxxx
HANDLER NAME	DMSUTL
DCS TRAN CODE	0000
BOJ STATUS (UP/DOWN)	UP
I/O MODE (FULL/PART)	FULL
AUDIT (YES/NO/PART)	NO
TRANSACTION TIME_OUT (SECONDS)	999
HANDLER-LINK WELCOME FORM NUM	0000
SPECIAL ROUTING (SO=SIGNON, LK=LINK, CP=CLEAR)	
TEST MODE (YES/NO)	NO
TRANSACTION TYPE (NDL/SRI/CHAIN)	SRI
INITIATOR BYPASS FORM INDEX	000
BREAKOUT NOTIFY (YES/NO)	NO
CHAIN NUMBER	0000
PRIORITY	00
PARSE START NUMBER	0000
PARSING LENGTH	00

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
General Utility Information

Table 1-3. HANDLER Definition

USER: xxxxxxxxx CP3680 RESIDENT HANDLER DATE: MM/DD/YY
 VIEW PROCESS nnnn FORM: xxxx

HANDLER NAME	DMSUTL	SCHEDULE	03
BOJ (UP/DOWN)	UP	(1=ROUND ROBIN, 2=NEXT AVAILABLE..)	
MODE (FULL/PART)	FULL	HAND=>HAND (YES/NO)	NO
PRIVILEGED HANDLER (YES/NO)	NO	QUE MSGS IF HANDLER IS DOWN	NO
SUPPRESS CONTINUATION (YES/NO)	NO	ONE IN ONE OUT (YES/NO)	NO
HANDLER WELCOME FORM NUMBER	0000	DOWN HARD-LINKED STNS IF	NO
1 2 3 4 5 6 7 8 10			

	1 - 10
HANDLER	11 - 20
FORM	21 - 30
NUMBERS	31 - 40
	41 - 50
	51 - 60
	61 - 70
	71 - 80
	81 - 90
	91 - 100
	101 - 110
	111 - 120
	121 - 130

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
General Utility Information

Table 1-4. PROCESS Definition

```

USER:  xxxxxxxxxxxx      CP3680 RESIDENT HANDLER      DATE:  MM/DD/YY
                        VIEW PROCESS nnnn              FORM:  xxxx

                        ROLL  PACK NAME  VALUE  MSG  HOST
                        ID  CG  OUT  IF ON PACK X=YYYYYY FLOW  PRI
PROCESS (PROG_ID) DMSUTL xxxxx 000   xxxxx           2    x

NAME OF HOST      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

HANDLER NAME      DMSUTL           DCS PRIORITY 0 (NOT IMPLEMENTED)

HEADER TYPE (DCS, DCP, MCS, NDL, USER,,, GMCS)          NDL

IS PROCESS ("UP", "DOWN") AT BEGINNING OF HANDLER        DOWN

UP WHEN THRESHOLD EXCEEDED ("YES", "NO")      YES

QUEUE THRESHOLD TO UP PROCESS  001 IDLE DOWN TIMEOUT 0000 SECONDS

MANUALLY EXECUTE PROCESS ("Y/N")      N

ALWAYS UPDATE CONTINUATION FOR NDL ("YES"/"NO")      YES

MCP-SECURITY:  USER CODE

PASSWORD

COM TYPE (CRCR, STOQ, USER, MSTQ)          STOQ

QUEUE LIMIT

SEND CRCR WAKEUP (Y/N)

CLEAR QUEUE (Y/N)

TO HOST

```

USING THE DMSUTL HELP SCREENS

A DMSUTL help screen displays information. You cannot enter, change, or delete information that appears on a help screen. The DMSUTL help screens are divided into three categories: general help information for each screen, DMSUTL help information for each utility, and freeform syntax help information with syntax diagrams.

You can move from field to field within a screen by using the TAB key to go forward, and the REVERSE TAB key to go backwards. The following procedures explain how to go to each type of help screen.

General Help Screen Procedure

- To go to a general help screen from any DMSUTL screen, move the cursor to the ACTION field or any area that is not a field and press the SPCFY key.
- To return to the DMSUTL screen you were using, press the SPCFY key.

DMSUTL Help Screen Procedure

- To go to a DMSUTL general help screen, place the cursor in the YOUR CHOICE IS field and press the SPCFY key.
- From the DMSUTL general help screen, you can go to a help screen for each specific utility. For example, to go to the help screen for DMSAUD, enter AUD in the ACTION field and press the TRANSMIT key. To go to help screens for other utilities, follow the directions on the DMSUTL general help screen.

Freeform Syntax Help Screen Procedure

- To go to the general freeform help screen, place the cursor in the FREEFORM SYNTAX field and press the SPCFY key.
- To go to a particular freeform syntax help screen, follow the directions on the general freeform help screen.

The following common choices appear on the help screens, as applicable: REFresh, MORE-help, PREvious-form, NEXt-form.

REF

The refresh option enables you to recover your screen if it is partially or fully rendered useless, or you are knocked out of FORMS mode. If you are still in FORMS mode, type REF in the ACTION field. If you are not in FORMS mode, press the HOME key, type REF, and press the TRANSMIT key.

MOR

The more-help option enables you to go to another help screen that contains more information about a particular subject. To use the more-help option, type MOR in the ACTION field and press the TRANSMIT key.

PRE

The previous-form option enables you to go back to the help screen you just used. To use the previous-form option, type PRE in the ACTION field and press the TRANSMIT key.

NEX

The next-form option enables you to go to the next help screen. To use the next-form option, type NEX in the ACTION field and press the TRANSMIT key.

DMSUTL permits you to enter one, two, or three letters for each choice. For example, you can enter R, RE, or REF to refresh your screen.

USING THE DMSUTL DATA ENTRY SCREENS

A DMSUTL data entry screen can display information; however, its main purpose is to collect information for the DMS utility (DMSUTL) program.

You can move from field to field within a screen by using the TAB key to go forward, and the REVERSE TAB key to go backwards. The following paragraphs explain the common screen choices for data entry screens and provide other information about the use of the screens. Detailed information for each screen appears later in this manual.

Common Screen Choices

The following common screen choices appear on each data entry screen: REFresh, HOME, and QUIt.

REF

The refresh option enables you to recover your screen if it is partially or fully rendered useless, or you are knocked out of FORMS mode. If you are still in FORMS mode, type REF in the ACTION field. If you are not in FORMS mode, press the HOME key, type REF, and press the TRANSMIT key.

HOM

The home option enables you to return to the main DMSUTL menu screen, which is called the home menu screen. You can go to this screen from any data entry screen. To use the home option, type HOM in the ACTION field and press the TRANSMIT key.

QUI

The quit option enables you to terminate the execution of the DMSUTL program (go to End Of Job) from any screen. Also, all necessary cleanup is done (such as, closing printer backup files, etc.). To use the quit option, type QUI in the ACTION field and press the TRANSMIT key.

DMSUTL permits you to enter one, two, or three letters for each choice. For example, you can enter Q, QU, or QUI to quit.



HOME MENU

```

ACTION:      HOME MENU                      10:23 A.M.      DMSUTL
            [          ]
            REFresh    QUIT                Press SPCFY for Help

```

```

UNISYS V SERIES DMSII UTILITIES
RELEASE 7.2

```

```

FUNCTIONS AVAILABLE:

```

1. DMSTAT	6. DDFLIS
2. DMSLST	7. DMSGEN
3. DMSCTL	8. DBPDMP
4. DMSAUD	9. DMSDLR
5. DMSDAP	

```

YOUR CHOICE IS: [ ]

```

```

DATABASE DESIRED: [          ] ON [          ] ISAM: [ ]

```

```

ENTER FREEFORM SYNTAX IF YOU DON'T WANT ANY MORE SCREENS:
[

```



1. REFresh
QUIt
For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of this section.
2. FUNCTIONS AVAILABLE: These are the various utility programs that you can access with the DMSUTL utility.
3. YOUR CHOICE IS: You enter a number from 1 to 9, corresponding to the utility you wish to use. Then, you will get a menu screen on which you can enter the parameters for that utility.
4. DATABASE DESIRED: Enter the database name, from three to six characters in length.
5. ON: Enter the location of the DDF file (XXXDDF). Valid entries are DISK, PACK, and the name <familyname> of a pack. If you enter PACK, the utility will look for the DDF on unrestricted packs. The default value is PACK.
6. ISAM: The DMSUTL program uses the ISAM field only for the DMSLST utility (choice 2) for the purpose of listing a nonDMS ISAM file. Do not check this field to list a database structure file that has an indexed sequential organization.
7. ENTER FREEFORM SYNTAX: Experienced users of DMSUTL and the DMS utilities may not want to traverse the tree of DMSUTL screens to use the desired functions. You can enter a syntax in this field that will begin the action you want with no further interface or prompting from DMSUTL.

For functions that you can initiate with the IR command, the syntax is identical to the IR command syntax, except that you do not enter "IR [CARD or SPO] <dbname>" as part of the syntax. You enter the database name and location of the control file in the fields of the home menu screen as for any other function, and then you enter the rest of the syntax in the ENTER FREEFORM SYNTAX field.

You can view the freeform syntax diagram for any function by pressing the SPCFY key. You will get the freeform syntax general help screen, which provides instructions for you to get the freeform syntax help screen for a particular utility.

USING DMSUTL IN BATCH MODE

You can access the DMSTAT, DMSLST, DMSCTL, DMSDAP, DDFLIS, and DMSGEN utilities by initiating DMSUTL with commands from the ODT, a card deck, or a Workflow Language (WFL) program. This batch mode method is useful for WFL users and for users who do not have access to a data communications terminal.

Executing DMSUTL from the ODT

Execute DMSUTL from the ODT with the EXECUTE MCP control instructions and with AX (ACCEPT) MCP control instructions, as follows:

```
EX DMSUTL/SPO
<mix number> AX "<parameters>"
<mix number> AX END
```

The "Parameters" portion of this section contains a discussion of <parameters>.

Executing DMSUTL from Cards or Card Images

To execute DMSUTL from a card deck or pseudodeck, the deck should look like this:

```
? EX DMSUTL/CARD
? DATA CARD
<parameters>
? END
```

The "Parameters" portion of this section contains a discussion of <parameters>.

For information on pseudodecks, refer to the LDCNTL utility and to the RN and CHANGE commands in the *V Series MCP/VS System Software Operations Guide Volume 3: System Utilities* and *Volume 2: System Commands* and to the COPY TO CTLD command in the *B 2000/B 3000/B 4000/V Series CANDE Installation and Operation Guide* or the *V Series CANDE Installation and Operations Guide*. The section entitled "About This Manual" tells which of these manuals applies to each operating system.

Initiating DMSUTL from WFL

To initiate DMSUTL in batch mode from a WFL program, use this WFL source format.

```
RUN DMSUTL (,CARD);
DATA CARD
<parameters>
!
```

The "Parameters" portion of this section contains a discussion of <parameters>.

PARAMETERS

The parameters you use in batch mode are quite similar, but not identical, to the freeform parameters you used in screen mode. The differences in batch mode parameters are:

- You must enter the database name at the beginning of the syntax, in the form DB = <database name>.
- If you use DMSLST to list an ISAM file, enter the ISAM data file name instead of <database name> and enter the word "ISAM" afterwards.
- Because there is no possibility of viewing from the terminal when you use batch mode, you cannot use the LIST, PRINT, or PRINTONLY options. Use RELEASE to end one printer backup file and start another.

Figures 1-2 and 1-3 show the parameter syntax for batch mode. The specifications for each function appear later in this manual where a description of each function appears. To get a copy of the batch mode syntax, enter the following at the ODT:

```
EX DMSUTL/SPO
<mix number> AX HELP
<mix number> AX END
```

< parameters >

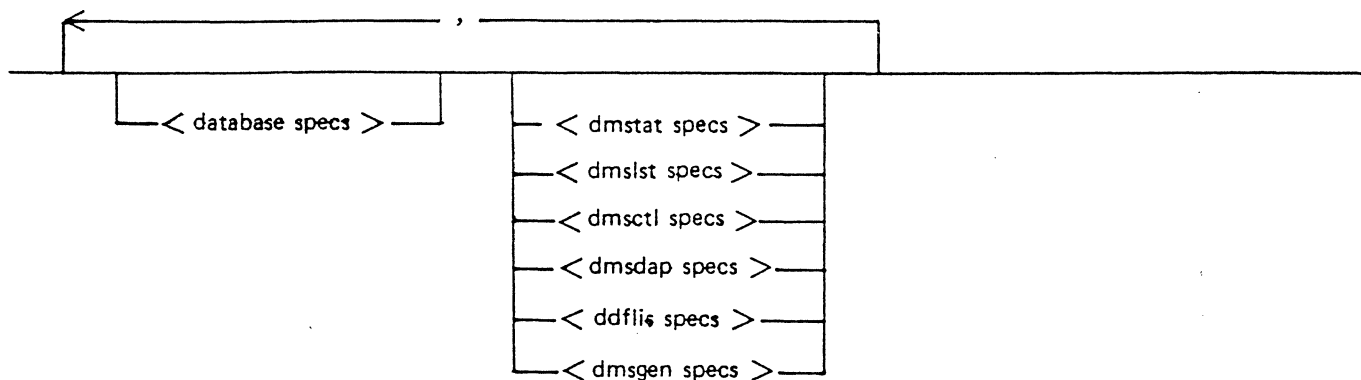


Figure 1-2. Parameters for Using DMSUTL in Batch Mode

< database specs >

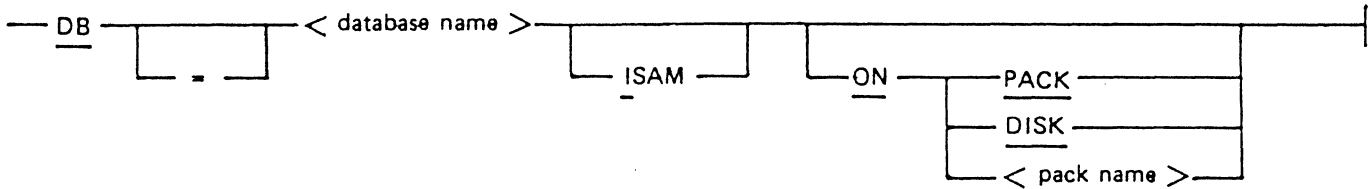


Figure 1-3. Database Specifications for Using DMSUTL in Batch Mode

<database name>

Database name is the name (3 to 6 characters) of the database. For ISAM files, use the data file name.

ISAM

When you execute DMSLST, enter ISAM to list ISAM files.

ON PACK

ON DISK

ON <pack name>

ON PACK/ON DISK/ON <pack name> represent the location of the Data Definition File (DDF) or the ISAM data file. The default value is PACK, which means any system resource pack.

<dmstat specs>

<dmslst specs>

<dmsctl specs>

<dmsdap specs>

<ddflis specs>

<dmsgen specs>

The sections on DMSTAT, DMSLST, DMSCTL, DMSDAP, DDFLIS, and DMSGEN include discussions of the specifications for each.

SECTION 2

DATABASE PROGRAM DUMP ANALYSIS UTILITY (DBPDMP)

INTRODUCTION

The database program (DBP) dump analysis utility provides Unisys technical personnel with information they need to resolve problems with the Data Management System II (DMSII). When the DBP fails and a dump is generated, the DBPDMP utility program can be used to read the dump file and produce a printer backup file containing a highly technical analysis of the state of the DBP. You should send the raw dump file (\$pnnnn) on a tape with any Field Communication Form (FCF) so that Unisys personnel can use this tool to analyze the problem. Also, you should include on the tape all the files necessary to reproduce the problem (if possible) so that the FCF will not be rejected if the dump proves to be inconclusive.

INITIATING DBPDMP - SCREEN MODE

To use DBPDMP with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. On the home menu screen enter an 8 in the YOUR CHOICE IS field, the database name in the DATABASE DESIRED field, and the location (DISK, PACK, or <diskpack name>) of the database control file in the ON field. Then, press the TRANSMIT key.

Figure 2-1 shows the screen that will appear on your terminal.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Database Program Dump Analysis Utility (DBPDMP)

```
ACTION:      DBPDMP MENU          10:23 A.M.      DMSUTL
              [ REFresh  HOME  QUIT ]
```

PROCESSOR NUMBER: []

DUMP FILE NUMBER: []

OPTIONS:

blank: ALL INTERNAL DBP TABLES, BUFFERS, ETC. PLUS RAW DBP MEMORY.
A : SAME AS <blank> PLUS ALPHA TRANSLATION OF RAW DBP MEMORY.
TB : ALL INTERNAL DBP TABLES, BUFFERS, ETC. (NO RAW DBP MEMORY).
GL : GLOBAL DBP TABLES ONLY (NO STRUCTURE INFORMATION).
TA : TASK INFORMATION ONLY.
AU : AUDIT INFORMATION ONLY.
RC : AUDIT AND RECOVERY INFORMATION ONLY.
ST : STRUCTURE INFORMATION ONLY.
RG : STRUCTURE AND REORGANIZE INFORMATION ONLY.
RW : RAW DBP MEMORY ONLY (NO ALPHA TRANSLATION).

YOUR CHOICE IS: []

Figure 2-1. DBPDMP Menu Screen

1. REFresh
HOME
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. PROCESSOR NUMBER: Enter the number of the processor on which the dump file was created. This is the number immediately after the \$ in the dump file name (\$pnnnn). Enter zero for nonshared systems.
3. DUMP FILE NUMBER: You may omit leading zeros. Do not include the \$ or the processor number. For example, if the file name of a dump file as it appears on the ODT is \$21234, enter 1234.

4. OPTIONS:

The option that you choose directs DBPDMP to create a report from the dump that provides details on the following:

- <blank> : All internal DBP tables, buffers, etc. plus the raw DBP memory.
- A : Same as <blank> plus alpha translation of the raw DBP memory.
- TB : All internal DBP tables, buffers, etc. (no raw DBP memory).
- GL : Global DBP tables only (no structure information).
- TA : Task information only.
- AU : Audit information only.
- RC : Audit and recovery information only.
- ST : Structure information only.
- RG : Structure and reorganize information only.
- RW : Raw DBP memory only (no alpha translation).

5. YOUR CHOICE IS: Enter <blank>, A, TB, GL, TA, AU, RC, ST, RG, or RW.

FREEFORM SYNTAX

The freeform syntax diagram for DBPDMP appears in Figure 2-2. You can enter this syntax on the bottom of the home menu screen. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

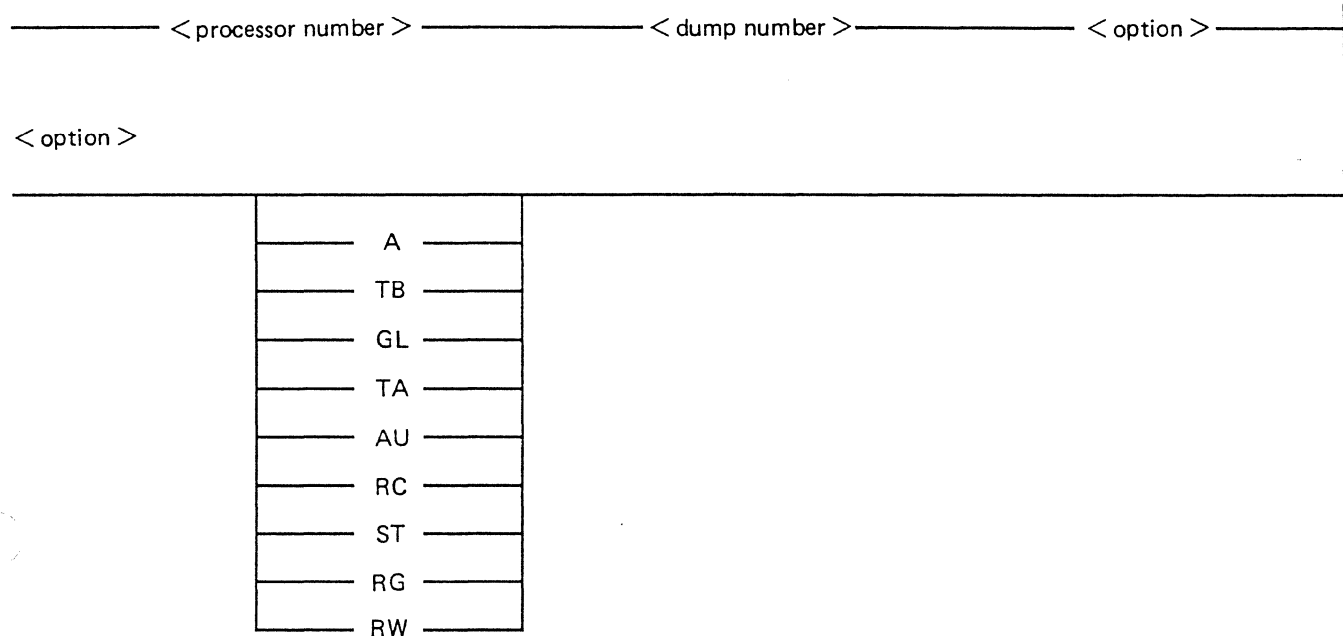


Figure 2-2. DMSUTL Freeform Syntax for DBPDMP

<processor number>

The processor number is a 1-digit number that indicates where the dump file was created. For example, \$21234 is a dump file name. The 2 following the \$ is the processor number.

<dump number>

When the a dump file name is \$21234, the dump number is: 1234.

See the OPTIONS portion of the DBPDMP menu screen description in this section for a discussion of each option that appears in the freeform syntax diagram.

INITIATING DBPDMP - STAND-ALONE MODE

You cannot initiate DBPDMP with the DMSUTL batch mode. However, because DBPDMP is an independent program, you can initiate it with the PM command, from the ODT, from cards or card images, or from a WFL job. In the following paragraphs, there are references to *pnnnn*. In these references, *p* represents the processor number on which the dump file was created and *nnnn* represents the dump number including any leading zeros.

With the PM Command

You can initiate the DBPDMP utility by entering the PM command with a dump file number that refers to a DBP dump file. The format for the PM command is:

PM *pnnnn*

For a complete discussion of the PM command, refer to the *V Series MCP/VS System Software Operations Guide, Volume 2: System Commands*.

From the ODT

You can execute DBPDMP by entering the following syntax at the ODT:

EXECUTE DBPDMP/*pnnnn*
or
EXECUTE DBPDMP/*pnnnn*/*<option>*

From Cards or Card Images

You can execute DBPDMP by entering the following syntax in a card deck or pseudodeck:

?EXECUTE DBPDMP/*pnnnn*
or
?EXECUTE DBPDMP/*pnnnn*/*<option>*

From a WFL Job

You can initiate DBPDMP with the following WFL statements:

RUN DBPDMP ("*pnnnn*")
or
RUN DBPDMP ("*pnnnn*" , "*<option>*")

EXAMPLES

Examples of ODT Commands

```
EXECUTE DBPDMP/20005  
EXECUTE DBPDMP/12  
EXECUTE DBPDMP/10022/TB
```

Examples of Card Images

```
?EXECUTE DBPDMP/00002  
?EXECUTE DBPDMP/22/A
```

Examples of WFL Statements

```
RUN DBPDMP ("12")  
RUN DBPDMP ("10022","GL")
```

SECTION 3

STATISTICS REPORTING UTILITY (DMSTAT)

INTRODUCTION

DMSTAT is a statistics utility that provides facilities for dealing with the information in a database's statistics file. It has two functions. First, it can input a statistics file, control file, and Data Definition File (DDF), and generate a display or a printer backup file containing statistical reports on a database. Second, it can clear (reinitialize) the data in a statistics file and get it ready to collect a new batch of statistics.

These functions apply only to a database that has its STATISTICS option set, because only such a database has a statistics file.

When you use the report function, you can obtain the following:

- A DATABASE REPORT that includes statistics accumulated for a database as a whole.
- A STRUCTURE REPORT that includes statistics accumulated for database structures. The structure report can contain information about each and every structure in a database with a set of totals for all structures, or it can include information about only one particular structure.
- A USER REPORT that includes statistics accumulated for each user program execution that opened the database.

You can request any or all of these reports at any time, but each report's statistics are updated at different time intervals (see the descriptions of each report). The DMSTAT program puts the reports into a printer backup file named DMSUTL/<database name>.

INITIATING DMSTAT - SCREEN MODE

To use DMSTAT with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 1 in the YOUR CHOICE IS field on the home menu screen. Put the name of the database in the DATABASE DESIRED field; put the location (DISK, PACK, or <diskpack name>) of the database in the ON field. Then, press the TRANSMIT key.

Figure 3-1 shows the screen that will appear on your terminal.

```

                                DMSTAT MENU                                10:23 A.M.      DMSUTL
ACTION:  [                                ]
          LIST  PRInt  print-ONLY  RELease-printer  REInitialize
          REFresh HOME  QUIT          Press SPCFY for Help
-----
REPORT OPTIONS:

    DATABASE REPORT  [ ]
    STRUCTURE REPORT [ ]
    USER REPORT     [ ]

SPECIFIC STRUCTURE  [                ]
-----
```

Figure 3-1. DMSTAT Menu Screen

1. REFresh
HOME
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. LIST: The DMSUTL program displays the output on the terminal only.
3. PRInt: The DMSUTL program displays the output on the terminal and writes it to a printer backup file.
4. print-ONLY: The DMSUTL program writes the information to a printer backup file, but does not display it on the terminal.

5. RELease: The DMSUTL program closes the printer backup file so that you can print the output.
6. REInitialize: When you enter REI in the ACTION field, the statistics file is cleared and on the next DBP execution the DBP puts the statistics in a clean file. If you do not enter REI, the DBP maintains statistics over successive runs of DMSTAT. When you enter REI, the DMSTAT program generates all three reports which enables you to archive the previous statistics.

In the ACTION field, enter the capitalized letters to indicate your choice of action. The default value is LISt.

REPORT OPTIONS:

You may request any combination of these report options.

7. DATABASE REPORT: When you check this field (with any nonblank character), you request a report on statistics for the database as a whole. A sample copy of this report appears in Figure 3-5.
8. STRUCTURE REPORT: When you check this field, you request a report on the statistics accumulated for database structures. You can use this with the SPECIFIC STRUCTURE field. If you leave the SPECIFIC STRUCTURE field blank, the DMSTAT program generates a report for all of the structures in the database. When you LISt this report, you can skip from one structure to the next by entering NEX in the ACTION field. You will also get the next structure report at the end of each structure report. You can view the totals for all structures by entering TOT in the ACTION field at the top of any report screen. A sample copy of this report appears in Figure 3-6.
9. USER REPORT: When you check this field, you request a report on the statistics accumulated for each user program execution that opened the database. A sample copy of this report appears in Figure 3-7.
10. SPECIFIC STRUCTURE: If you check the STRUCTURE REPORT field and fill in the SPECIFIC STRUCTURE field, the DMSTAT program will report on statistics only for the specified structure. If you check the STRUCTURE REPORT field and leave the SPECIFIC STRUCTURE field blank, the DMSTAT program reports on all of the structures in the database. Whether or not you fill in the SPECIFIC STRUCTURE field, the DMSTAT program generates totals for all structures and you may view them by entering TOT in the ACTION field at the top of any report screen. A sample copy of this report appears in Figure 3-6.

The DMSTAT program checks the timestamps in the statistics file (XXX001) and the control file. If the timestamps do not match, the DMSTAT program returns an error message to the terminal. To override this mismatch, re-transmit the screen. This capability is useful for printing old statistics files.

FREEFORM SYNTAX

The freeform syntax for DMSTAT appears in Figure 3-2. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSTAT menu screen. For further discussion of the options, refer to the "Initiating DMSTAT - Batch Mode" portion of this section. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

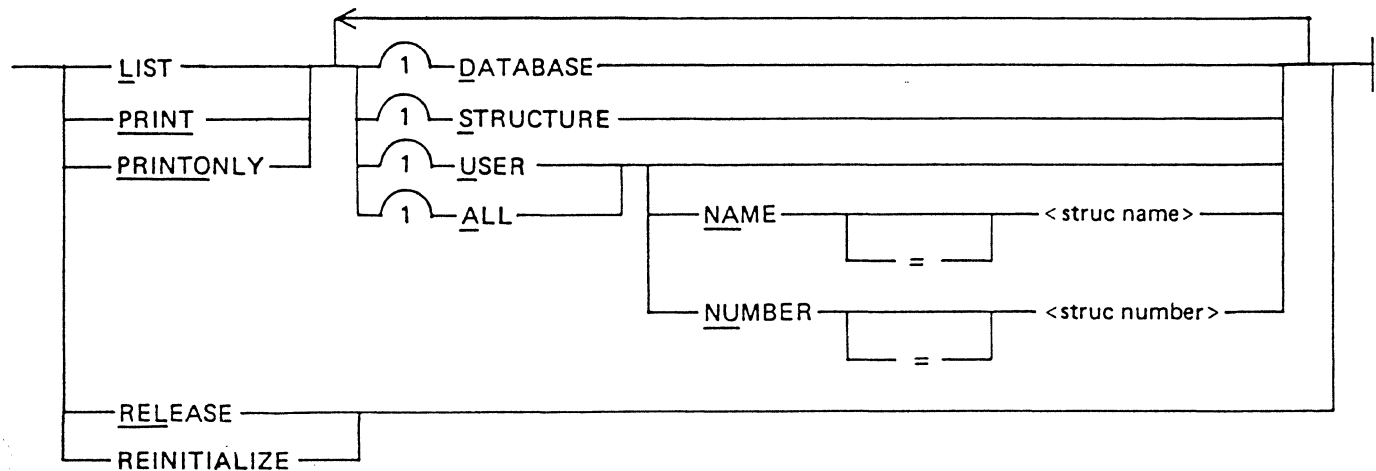


Figure 3-2. Freeform Syntax for DMSTAT

INITIATING DMSTAT - BATCH MODE

For general directions on use of the batch mode, refer to the "Using DMSUTL in Batch Mode" portion of Section 1. Figure 1-3 shows the general syntax; Figures 3-3 and 3-4 show the DMSTAT specification syntax.

< dmstat specs >

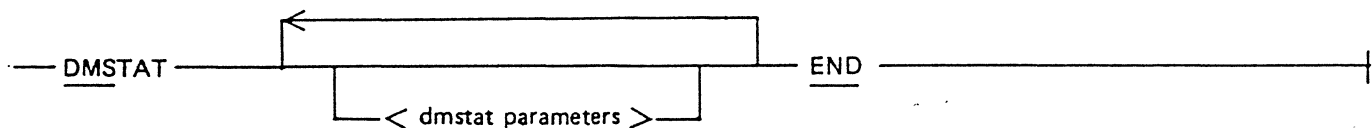


Figure 3-3. DMSTAT Specification for Batch Use

<dmstat parameters>

You must enter <dmstat parameters> on a single line of input; they cannot carry over to two cards or two AX messages. For more information about the options, refer to the "DMSTAT Report Options" portion of this section.

<dmstat parameters>

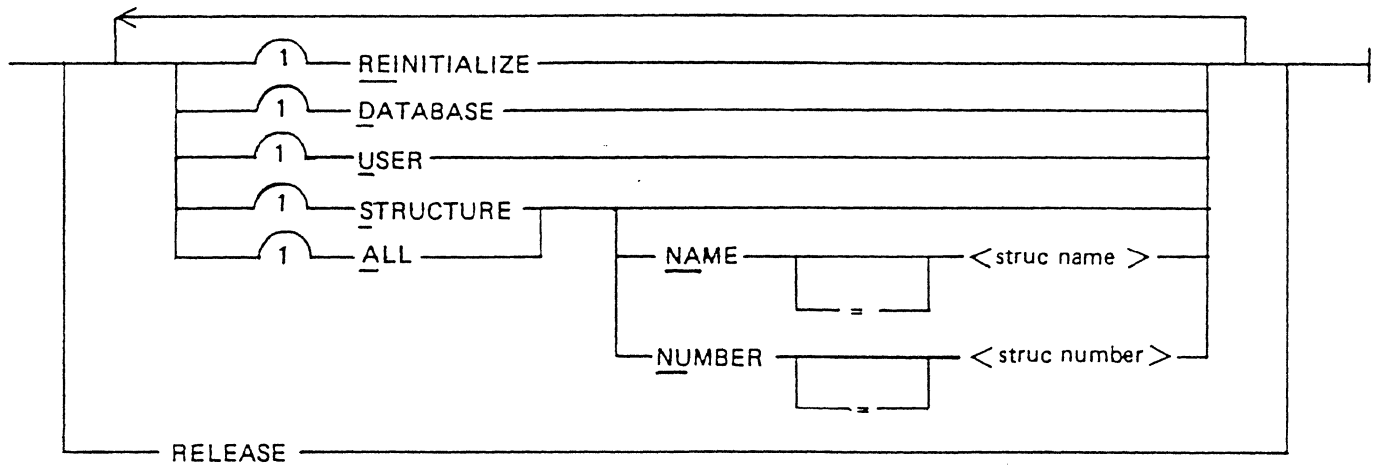


Figure 3-4. DMSTAT Parameters for Batch Use

REINITIALIZE

With this parameter, the DMSTAT program reinitializes all statistics to zero. When you use this option, the DMSTAT program will automatically generate the database, structure, and user reports which enables you to archive the previous statistics.

DATABASE

With this parameter, the DMSTAT program creates a report of the statistics accumulated by the database as a whole.

USER

With this parameter, the DMSTAT program creates a report showing the statistics for each user program execution that opened the database.

STRUCTURE

With this parameter, the DMSTAT program creates a report showing the statistics accumulated by the database program (DBP) for each structure. If you want statistics for only one particular structure, use the NAME or NUMBER options.

ALL

With this parameter, the DMSTAT program creates the database, structure, and user reports. You can use the NAME and NUMBER options to limit the structure report to one structure.

<struc name>

The structure name has a maximum length of 17 characters.

<struc number>

The structure number has a maximum length of 4 digits.

RELEASE

With this parameter, the DMSTAT program closes the current printer backup file so that you can print it.

DMSTAT REPORT OPTIONS

The DMSTAT menu screen offers these report options: database report, structure report, and user report. The following paragraphs cover each of these report options.

Database Report

The database report provides those statistics that the DBP has accumulated for the database as a whole since the last reinitialization. The data for this report is updated every time snapshot statistics are taken (every time a status call is received from the MCP) and when the database is closed. A sample copy of this report appears in Figure 3-5 and definitions of the terms from the report appear in the following paragraphs.

UNISYS B2000/B3000/B4000/V SERIES DMSII STATISTICS REPORT FOR ADMIN ON PACK

DATE: 07/21/87 TIME: 11:03

DATABASE REPORT

TOTAL TIME DATABASE OPENED (HHHH:MM:SS) 0000:09:30

DATABASE OPENS - TOTAL	10	USER TERMINATES	9
UPDATE	8	DEADLOCKS	0
DYNAMIC SIZE - MAXIMUM	176000	BUFFERS RETURNED	151
MINIMUM	119000	EXPAND CORE REQUESTS	2
CURRENT	176000	RETURN CORE REQUESTS	0
NUMBER OF USERS - MAXIMUM	11	STATUS CALLS	3
CURRENT	0		

STATISTICS FROM LAST STATUS CALL

USERS - TOTAL	11	USERS WAITING - LOCK	3
ACTIVE	3	BLOCK	0
		BLOCK LOCK	0
DYNAMIC SIZE - CURRENT	176536	FREE LIST	0
IN-USE	155596	OPEN/CLOSE	0
AVAILABLE MEMORY AREAS	61	RESTART	0

Figure 3-5. Sample of DMSTAT Database Report

TOTAL TIME DATABASE OPENED

Total amount of time that one or more user programs had the database open.

DATABASE OPENS - TOTAL

Number of times the database was opened by a user program.

DATABASE OPENS - UPDATE

Number of times the database was opened UPDATE by a user program.

USER TERMINATES

Number of times a user program was terminated while it was using the database, or when it went to End Of Job (EOJ) without explicitly closing the database.

DEADLOCKS

Number of deadlocks that were detected (and restarted) by the DBP.

DYNAMIC SIZE - MAXIMUM

Maximum amount of dynamic memory allocated in the DBP at any one time. You should set ALLOWEDCORE to a value slightly higher than this number.

DYNAMIC SIZE - MINIMUM

Amount of dynamic memory allocated in the DBP just after initialization. This number should reflect the STACKSIZE value of the database.

DYNAMIC SIZE - CURRENT

Amount of dynamic memory allocated in the DBP at snapshot statistics time.

NUMBER OF USERS - MAXIMUM

Maximum number of user programs that had the database open at any one time.

NUMBER OF USERS - CURRENT

Number of user programs that had the database open at snapshot statistics time.

BUFFERS RETURNED

Number of infrequently used buffers in the DBP dynamic memory area that the DBP has reclaimed for free space while servicing status calls.

EXPAND CORE REQUESTS

Number of times the DBP has acquired more dynamic memory from the operating system (5 + KD for each request).

RETURN CORE REQUESTS

Number of times the DBP has returned dynamic memory no longer in use (free space) to the operating system (minimum return of 4 KD).

STATUS CALLS

Number of times the operating system has requested that the DBP perform a status function. When the DBP performs a status function, it does two things. First, it takes a statistical snapshot of itself (database report statistics only) and writes it out to the statistics file. Then, it returns infrequently used memory within its dynamic memory area to the operating system (by means of a RETURN CORE REQUEST).

USERS - TOTAL

Number of user programs that had the database open at snapshot statistics time.

USERS - ACTIVE

Number of user programs that had a database request outstanding at snapshot statistics time.

USERS WAITING LOCK

Number of user programs that were WAITING LOCK at snapshot statistics time.

USERS WAITING BLOCK

Number of user programs that were WAITING BLOCK at snapshot statistics time.

USERS WAITING BLOCK LOCK

Number of user programs that were WAITING BLOCK LOCK at snapshot statistics time.

USERS WAITING FREE LIST

Number of user programs that were WAITING FREE LIST at snapshot statistics time.

USERS WAITING OPEN/CLOSE

Number of user programs that were WAITING OPEN or CLOSE at snapshot statistics time.

USERS WAITING RESTART

Number of user programs that were WAITING FUNCTION RESTART at snapshot statistics time.

DYNAMIC SIZE - CURRENT

Amount of dynamic memory actually allocated in the DBP at snapshot statistics time.

DYNAMIC SIZE - IN-USE

Amount of dynamic memory actually in use in the DBP at snapshot statistics time.

AVAILABLE MEMORY AREAS

Number of dynamic memory pieces available for use in the DBP at snapshot statistics time.

Structure Report

The structure report provides a report of those statistics accumulated by the DBP for one structure or each structure in the database, followed by a report giving the totals for all of the structures. Note that the statistical data for each structure is updated every time that structure is physically closed, and all of the structure statistics are cleared at the time of reinitialization. A sample copy of a structure report appears in Figure 3-6 and definitions of the terms from the report appear in the following paragraphs.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Statistics Reporting Utility (DMSTAT)

STRUCTURE REPORT

RESTARTAREA	3	RESTART DATA SET		
READS - PHYSICAL	16	OPENS - UPDATE		1
LOGICAL	164	PHYSICAL		1
WRITES - PHYSICAL	23	LOGICAL		10
LOGICAL	154			
FINDS	0			
LOCKS	6			
DELETES	1			
LOCK/STORES	144			
CREATE/STORES	0			
BLOCK FOUND IN MEMORY	148	MAXIMUM BUFFERS		2
BUFFERS ADDED	11			
BUFFERS REUSED - IDLE	2			
CHANGED	3			
READAHEAD		RESET	WRITEAHEADS	0
			WRITEAHEAD WAIT IOC	0

RESTARTSET	4	INDEX SEQUENTIAL SET		
READS - PHYSICAL	16	OPENS - UPDATE		1
LOGICAL	22	PHYSICAL		1
WRITES - PHYSICAL	10	LOGICAL		10
LOGICAL	10			
FINDS - RANDOM	6			
SEQUENTIAL	0			
FIND KEY OF	0			
INSERTS	0			
REMOVES	1			
BLOCK SPLITS	0			
BLOCK FOUND IN MEMORY	6	MAXIMUM BUFFERS		2
BUFFERS ADDED	16			
BUFFERS REUSED - IDLE	0			
CHANGED	0			
READAHEAD		RESET	WRITEAHEADS	0
			WRITEAHEAD WAIT IOC	0

Figure 3-6. Sample of DMSTAT Structure Report (Sheet 1 of 5)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Statistics Reporting Utility (DMSTAT)

STRUCTURE REPORT				
EMPLOYEES	5	STANDARD DATA SET		
READS - PHYSICAL	39	OPENS - UPDATE		1
LOGICAL	1590	PHYSICAL		1
WRITES - PHYSICAL	42	LOGICAL		10
LOGICAL	246			
FINDS	1037			
LOCKS	225			
DELETES	0			
LOCK/STORES	154			
CREATE/STORES	30			
BLOCK FOUND IN MEMORY	1499	MAXIMUM BUFFERS		5
BUFFERS ADDED	31			
BUFFERS REUSED - IDLE	51			
CHANGED	9			
READAHEADS - PHYSICAL	0	WRITEAHEADS		12
LOGICAL	0			
WAIT IOC	0	WRITEAHEAD WAIT IOC		4
EMP-JOB-SET	6	UNORDERED LIST SUBSET		
** STRUCTURE OPENED BUT NOT ACCESSED **				
READS - PHYSICAL	11	OPENS - UPDATE		1
LOGICAL	11	PHYSICAL		1
WRITES - PHYSICAL	9	LOGICAL		10
LOGICAL	9			
DEPENDENTS	7	UNORDERED DATA SET		
READS - PHYSICAL	29	OPENS - UPDATE		1
LOGICAL	35	PHYSICAL		1
WRITES - PHYSICAL	56	LOGICAL		10
LOGICAL	33			
FINDS	0			
LOCKS	0			
DELETES	0			
LOCK/STORES	0			
CREATE/STORES	65			
BLOCK FOUND IN MEMORY	31	MAXIMUM BUFFERS		1
BUFFERS ADDED	12			
BUFFERS REUSED - IDLE	14			
CHANGED	28			
READAHEAD	RESET	WRITEAHEADS		0
		WRITEAHEAD WAIT IOC		0

Figure 3-6. Sample of DMSTAT Structure Report (Sheet 2 of 5)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Statistics Reporting Utility (DMSTAT)

STRUCTURE REPORT				
DEP-SET	8	ORDERED LIST SET		
READS - PHYSICAL	19	OPENS - UPDATE	1	
LOGICAL	110	PHYSICAL	1	
WRITES - PHYSICAL	31	LOGICAL	10	
LOGICAL	108			
FINDS - RANDOM	0			
SEQUENTIAL	0			
FIND KEY OF	0			
INSERTS	65			
REMOVES	0			
BLOCK SPLITS	0			
BLOCK FOUND IN MEMORY	66	MAXIMUM BUFFERS	3	
BUFFERS ADDED	18			
BUFFERS REUSED - IDLE	26			
CHANGED	0			
READAHEAD	RESET	WRITEAHEADS	21	
		WRITEAHEAD WAIT IOC	0	
EMP-NBR-SET	9	INDEX SEQUENTIAL SET		
READS - PHYSICAL	15	OPENS - UPDATE	1	
LOGICAL	108	PHYSICAL	1	
WRITES - PHYSICAL	15	LOGICAL	10	
LOGICAL	39			
FINDS - RANDOM	67			
SEQUENTIAL	0			
FIND KEY OF	0			
INSERTS	30			
REMOVES	0			
BLOCK SPLITS	0			
BLOCK FOUND IN MEMORY	93	MAXIMUM BUFFERS	2	
BUFFERS ADDED	15			
BUFFERS REUSED - IDLE	0			
CHANGED	0			
READAHEADS - PHYSICAL	0	WRITEAHEADS	0	
LOGICAL	0			
WAIT IOC	0	WRITEAHEAD WAIT IOC	0	

Figure 3-6. Sample of DMSTAT Structure Report (Sheet 3 of 5)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Statistics Reporting Utility (DMSTAT)

```

                                STRUCTURE REPORT

EMP-NAME-SET          10      INDEX SEQUENTIAL SET

    READS  - PHYSICAL          30      OPENS - UPDATE          1
             LOGICAL          1296      PHYSICAL              1
    WRITES - PHYSICAL          13      LOGICAL              10
             LOGICAL          39

    FINDS - RANDOM              0
             SEQUENTIAL        1195
             FIND KEY OF        0
    INSERTS                    30
    REMOVES                     0
    BLOCK SPLITS                0

    BLOCK FOUND IN MEMORY      1266      MAXIMUM BUFFERS          3
    BUFFERS ADDED              26
    BUFFERS REUSED - IDLE      4
                        CHANGED  0

    READAHEADS - PHYSICAL      0      WRITEAHEADS              0
                  LOGICAL      0
                  WAIT IOC      0      WRITEAHEAD WAIT IOC      0

ADDRESSES              11      RANDOM DATA SET

    READS  - PHYSICAL          38      OPENS - UPDATE          1
             LOGICAL          39      PHYSICAL              1
    WRITES - PHYSICAL          38      LOGICAL              4
             LOGICAL          39

    FINDS                    0
    LOCKS                    0
    DELETES                  0
    LOCK/STORES              0
    CREATE/STORES            34

    BLOCK FOUND IN MEMORY      1      MAXIMUM BUFFERS          2
    BUFFERS ADDED              10
    BUFFERS REUSED - IDLE      9
                        CHANGED  19

    READAHEAD                RESET      WRITEAHEAD                RESET

```

```

ADDR-SET          12      RANDOM DATA SET ACCESS
** STRUCTURE OPENED BUT NOT ACCESSED **

    OPENS                    4

```

Figure 3-6. Sample of DMSTAT Structure Report (Sheet 4 of 5)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Statistics Reporting Utility (DMSTAT)

STRUCTURE TOTALS				
STRUCTURES - TOTAL		17		
OPENED		15		
READS - PHYSICAL		289	OPENS - UPDATE	13
LOGICAL		3486	PHYSICAL	13
WRITES - PHYSICAL		283	LOGICAL	108
LOGICAL		788		
FINDS		1037	FINDS - RANDOM	73
LOCKS		231	SEQUENTIAL	1195
DELETES		1	FIND KEY OF	0
LOCK/STORES		298	INSERTS	149
CREATE/STORES		144	REMOVES	1
			BLOCK SPLITS	0
BLOCK FOUND IN MEM		3142	MAXIMUM BUFFERS	29
BUFFERS ADDED		179		
BUFFERS REUSED -IDLE		106		
CHANGED		59		
READAHEADS -PHYSICAL		0	WRITEAHEADS	33
LOGICAL		0		
WAIT IOC		0	WRITEAHEAD WAIT IOC	4

Figure 3-6. Sample of DMSTAT Structure Report (Sheet 5 of 5)

READS - PHYSICAL

Number of times the DBP initiated a physical read (I/O) on this structure file to retrieve a requested block. This number does not include any READAHEADS that have occurred, but it does include reads of Block one.

READS - LOGICAL

Number of times the DBP requested (or accessed) a block from this structure file.

WRITES - PHYSICAL

Number of times the DBP initiated physical writes (I/Os) on this structure file for one of the following reasons:

- To reuse a changed buffer
- To perform a controlpoint buffer flush
- To update Block one

The total number of actual writes is the number of physical writes plus the number of WRITEAHEADS.

WRITES - LOGICAL

Number of times that a user program function request changed a block from this structure file in memory.

OPENS - UPDATE

Number of times that a user program opened this structure with UPDATE.

OPENS - PHYSICAL

Number of times that the operating system physically opened this structure.

OPENS - LOGICAL

Number of times that a user program opened this structure.

FINDS

Number of FIND functions that user programs requested for data set records from this structure.

LOCKS

Number of times that user programs locked data set records in this structure.

DELETES

Number of data set records that user programs deleted from this structure.

LOCK/STORES

Number of data set records that user programs updated in this structure. In a restart data set, the DBP increments this number each time a user program makes an update with BEGIN-TRANSACTION AUDIT or END-TRANSACTION AUDIT, but it does not increment the number of LOCK operations. In some situations, this causes the number of LOCK/STORE operations to be larger than the number of LOCK operations.

CREATE/STORES

Number of new data set records that user programs added to this structure.

FINDS - RANDOM

Number of random FIND, LOCK or DELETE functions that user programs requested on this set.

FINDS - SEQUENTIAL

Number of sequential FIND, LOCK or DELETE functions that user programs requested on this set using NEXT or PRIOR syntax.

FINDS - FIND KEY OF

Number of FIND functions that user programs requested on this set using FIND KEY OF syntax.

INSERTS

Number of references that the DBP added to this indexed sequential structure while servicing user program STORE or INSERT function requests.

REMOVES

Number of references that the DBP removed from this indexed sequential structure while servicing user program DELETE or REMOVE function requests.

BLOCK SPLITS

Number of times that the DBP split an indexed sequential table of this structure. Note that the values of LOADFACTOR and BLOCKSIZE affect the number of block splits that occur.

BLOCK FOUND IN MEMORY

Number of times that a requested block was found to be already present in this structure's buffer pool during a logical read (that is, a physical read was not necessary).

BUFFERS ADDED

Number of times the DBP allocated a new buffer for this structure's buffer pool.

BUFFERS REUSED - IDLE

Number of times the DBP reused an idle buffer for a requested block from this structure. This number also includes the BUFFERS ADDED, READAHEADS - WAIT IOC, and the WRITEAHEADS WAIT IOC numbers that appear elsewhere on the report.

BUFFERS REUSED - CHANGED

Number of times the DBP reused a changed buffer for a requested block from this structure. Reusing a changed buffer involves writing the changed block before the new block can be read in. In some cases, it also involves an extra write to the audit trail.

MAXIMUM BUFFERS

Maximum number of buffers in this structure's buffer pool since the last time the statistics file was reinitialized.

READAHEADS - PHYSICAL

Number of times that the READAHEAD function caused an I/O to be initiated on this structure to get a requested block. READAHEAD allows the DBP to read blocks before they are needed which reduces the overall DBP I/O wait time.

READAHEADS - LOGICAL

Number of times that the READAHEAD function was initiated on this structure. In some situations, the number of logical READAHEADS may be greater than the number of physical READAHEADS. This happens because of a lack of idle buffers or because the READAHEAD block is already in memory.

READAHEADS - WAIT IOC

Number of times the DBP requested a block when there was already a READAHEAD in progress on that block. In these cases, the DBP waits for the I/O to complete. In some cases, increasing BLOCKSIZE can reduce this number.

WRITEAHEADS

Number of times the DBP wrote a changed buffer to this structure before that buffer was needed by another block request. Writing buffers in this way reduces overall DBP I/O wait time.

WRITEAHEADS WAIT IOC

Number of times that no idle buffer was available for reuse, but a WRITEAHEAD was already in progress for a changed buffer. It takes less time to wait for the WRITEAHEAD to complete than to force another buffer to be written. This number also includes the number of times that a user program requested a block when it already had a WRITEAHEAD in progress on the block.

User Report

The user report provides statistics on each user program that has opened the database since the last reinitialization. The DMSTAT program will provide these statistics for each user program execution. Note that the data for each user program is updated every time that user program closes the database. A sample copy of this report appears in Figure 3-7. Definitions of the terms from the report appear in the following paragraphs.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Statistics Reporting Utility (DMSTAT)

USER REPORT

ADMLOD

END DATE	06/27/86	END TIME	16:32:49
BEGIN DATE	06/27/86	BEGIN TIME	16:23:42
WAITING - LOCK	2	WAITING - FREE LIST	0
BLOCK	3	SYNCPOINT	0
BLOCK LOCK	0	RESTART	0
READS - PHYSICAL	87	OPENS - PHYSICAL	13
LOGICAL	394	LOGICAL	15
WRITES - PHYSICAL	107	CLOSES - PHYSICAL	0
LOGICAL	259	LOGICAL	15
FINDS	0	INSERTS - AUTOMATIC	55
LOCKS	30	MANUAL	0
DELETES	1	REMOVES - AUTOMATIC	1
LOCK/STORES	62	MANUAL	0
CREATE/STORES	62		
TRANSACTIONS	63	EXCEPTIONS	1
READAHEADS - PHYSICAL	0	WRITEAHEADS	10
LOGICAL	0		
WAIT IOC	0	WRITEAHEAD WAIT IOC	0

ADMLOD

END DATE	06/27/86	END TIME	16:32:57
BEGIN DATE	06/27/86	BEGIN TIME	16:27:39
WAITING - LOCK	2	WAITING - FREE LIST	0
BLOCK	0	SYNCPOINT	0
BLOCK LOCK	0	RESTART	1
READS - PHYSICAL	27	OPENS - PHYSICAL	0
LOGICAL	50	LOGICAL	15
WRITES - PHYSICAL	47	CLOSES - PHYSICAL	0
LOGICAL	42	LOGICAL	15
FINDS	0	INSERTS - AUTOMATIC	0
LOCKS	2	MANUAL	0
DELETES	0	REMOVES - AUTOMATIC	0
LOCK/STORES	0	MANUAL	0
CREATE/STORES	0		
TRANSACTIONS	0	EXCEPTIONS	2
READAHEADS - PHYSICAL	0	WRITEAHEADS	0
LOGICAL	0		
WAIT IOC	0	WRITEAHEAD WAIT IOC	0

Figure 3-7. Sample of DMSTAT User Report

BEGIN DATE BEGIN TIME

The date and time that this user program opened the database.

END DATE END TIME

The date and time that this user program closed the database.

WAITING - LOCK

Number of times the DBP placed this user program in the WAITING LOCK list.

WAITING - BLOCK

Number of times the DBP placed this user program in the WAITING BLOCK list.

WAITING - BLOCK LOCK

Number of times the DBP placed this user program in the WAITING BLOCK LOCK list.

WAITING - FREE LIST

Number of times the DBP placed this user program in the WAITING FREE LIST. The FREE LIST is a list of blocks containing available space.

WAITING - RESTART

Number of times the DBP placed this user program in the waiting FUNCTION RESTART list.

WAITING - SYNCPOINT

Number of times the DBP placed this user program in the WAITING SYNCPOINT list.

READS - PHYSICAL

Number of times the DBP initiated a physical read (I/O) to retrieve a requested block while servicing a request from this user program. This figure includes updates to a structure's Block one. The number of physical reads plus the number of physical READAHEADS gives the total number of reads.

READS - LOGICAL

Number of times the DBP requested (or accessed) a block while servicing a request from this user program.

WRITES - PHYSICAL

Number of times that the DBP initiated a physical write (I/O) while servicing a request from this user program. Physical writes occur in one of the following situations:

- When a changed buffer needs to be reused for another block request.
- When a controlpoint buffer flush occurs.

This figure includes writes to a structure's Block one. The number of physical writes plus the number of WRITEAHEADS gives the total number of actual writes.

WRITES - LOGICAL

Number of times that the DBP changed a block in memory while servicing a request from this user program.

OPENS - PHYSICAL

Number of structures that this user program physically opened. A physical open requires the involvement of the operating system. This implies that the user program was the first user of the structure.

OPENS - LOGICAL

Number of structures that this user program has opened.

CLOSES - PHYSICAL

Number of structures that had to be physically closed when this user program closed the database. A physical close requires the involvement of the operating system and is done only when there are no other user programs accessing the structure.

CLOSES - LOGICAL

Number of structures that this user program has closed.

FINDS

Number of times that this user program issued a successful FIND request, and thereby retrieved a data set record.

LOCKS

Number of times that this user program issued a successful LOCK request, and thereby locked a data set record.

DELETES

Number of times that this user program issued a successful DELETE request, and thereby deleted a data set record.

LOCK/STORES

Number of times that this user program issued a successful LOCK/STORE request sequence, and thereby updated an existing data set record. This number also includes all updates to restart data set records.

CREATE/STORES

Number of times that this user program issued a successful CREATE/STORE request sequence, and thereby created an existing data set record.

TRANSACTIONS

Number of times that this user program issued a successful BEGIN/END TRANSACTION request sequence, thereby completing a transaction against the database.

INSERTS - AUTOMATIC

Number of times that the DBP inserted references to a data set record into automatic sets or subsets while servicing a data set update request from this user program.

INSERTS - MANUAL

Number of times that this user program issued a successful INSERT request, and thereby added a reference to an existing data set record into a manual subset.

REMOVES - AUTOMATIC

Number of times that the DBP removed references to a data set record from automatic sets or subsets while servicing a data set update request from this user program.

REMOVES - MANUAL

Number of times that this user program issued a successful REMOVE request, and thereby removed a reference to a data set record from a manual subset.

EXCEPTIONS

Number of times that this user program received an exception back on a database function request.

READAHEADS - PHYSICAL

Number of times that the DBP initiated a READAHEAD I/O (to retrieve a block presumed to be needed by future requests by this program) while servicing a function request from this user program.

READAHEADS - LOGICAL

Number of times that the DBP initiated a READAHEAD while servicing a function request from this user program. In some situations, the number of logical READAHEADS may be greater than the number of physical READAHEADS. This happens because of a lack of idle buffers or because the READAHEAD block is already in memory.

READAHEADS - WAIT IOC

Number of times the DBP requested a block while servicing a request from this user program and a READAHEAD was already in progress on that block. In these cases, the DBP waits for the I/O to complete. In some cases, increasing BLOCKSIZE can reduce this number.

WRITEAHEADS

Number of times the DBP wrote a changed buffer to a structure opened by this user program before the buffer was needed by another block request. Writing buffers in this way reduces the overall DBP I/O wait time.

WRITEAHEADS WAIT IOC

Number of times that this user program waited pending completion of a WRITEAHEAD I/O. When no idle buffer is available for reuse, but a WRITEAHEAD is already in progress on a changed buffer, it is more efficient to wait for the WRITEAHEAD to complete than to force another buffer to be written. This process may cause still another WRITEAHEAD to be initiated before returning control to the user program. This number also includes the number of times the user program requested a block when it already had a WRITEAHEAD in progress on the block.

SECTION 4

STRUCTURE LISTING UTILITY (DMSLST)

INTRODUCTION

DMSLST has two functions:

- Listing database structure files

The DMSLST program will input the Data Definition File (DDF) and specified structure files and prepare a printer backup file containing your choice of various kinds of listings of the contents of database structure files.

- Listing Indexed Sequential-Access Method (ISAM) files

The DMSLST program will input the ISAM file that you specified and its associated key files. Also, it will prepare a printer backup file containing your choice of various kinds of listings of the contents of the ISAM file and of its key files.

The following paragraphs include discussions of the two DMSLST functions.

The DMSLST program names the printer backup file it generates DMSUTL/<database name> or, for an ISAM file, DMSUTL/<ISAM file name>.

LISTING DATABASE FILES - SCREEN MODE

To use DMSLST with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 2 in the YOUR CHOICE IS field on the home menu screen. Put the name of the database in the DATABASE DESIRED field; put the location (DISK, PACK, or <diskpack name>) of the control file and DDF in the ON field. The ISAM field is only for listing non-database ISAM files; do not check it even if your database file happens to have an indexed sequential organization. Then, press the TRANSMIT key.

Figure 4-1 shows the screen that will appear on your terminal.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

```

ACTION:  DMSLST MENU                                10:23 A.M.      DMSUTL
         [ List Print print-ONLY RElease-printer
           REfresh HOME QUIT                          Press SPCFY for Help

```

REPORT OPTIONS:

```

CONTROL      [ ]
BLOCK        [ ]
RECORD       [ ]
BCI          [ ]
BLOCK ONE    [ ]
SUMMARY      [ ]

```

SPECIFIC STRUCTURE []

BLOCK RANGE: [] THROUGH []

Figure 4-1. DMSLST Menu Screen

1. REfresh
HOME
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. LISt: The DMSUTL program displays the output on the terminal only.
3. PRInt: The DMSUTL program displays the output on the terminal and writes it to a printer backup file.
4. print-ONLY: The DMSUTL program writes the information to a printer backup file, but does not display it on the terminal.
5. RElease: The DMSUTL program closes the printer backup file so that you can print the output.

In the ACTION field, enter the capitalized letters to indicate your choice of action. The default value is LISt.

Then, check the field (with any non-blank character) opposite any of the following report options:

6. CONTROL: The DMSLST program lists/prints the contents of the database control file (XXXCNF). A sample copy of this report appears in Figure 4-8.
7. BLOCK: The DMSLST program will list/print the entire contents of the block(s) that you specify. Use the "Block Range" fields to select specific block ranges. If you do not specify a block range, the DMSLST program lists all blocks. A sample copy of this report appears in Figure 4-9.
8. RECORD: The DMSLST program lists/prints the contents of the block(s) you specify and formats them into records. You can select a block range, if desired. If you do not specify a block range, the DMSLST program lists all blocks. A sample copy of this report appears in Figure 4-10.
9. BCI: The DMSLST program will list/print Block Control Information (BCI) for the block(s) you specify. You can select a block range, if desired. If you do not specify a block range, the DMSLST program lists the BCI for all of the blocks. A sample copy of this report appears in Figure 4-11.
10. BLOCK ONE: The DMSLST program lists/prints the first block (file header) for the structure. A sample copy of this report appears in Figure 4-12.
11. SUMMARY: The DMSLST program lists/prints a brief summary of the structure you specify. A sample copy of this report appears in Figure 4-13.
12. SPECIFIC STRUCTURE: Enter a structure name, or a structure number from 1 through 999.
13. BLOCK RANGE: Use these fields to select specific ranges of blocks. To see all blocks in the structure, leave both fields blank. To see only one block, enter a block number (1 to 8 digits) in the first field. To see a range of blocks, fill in both fields.

FREEFORM SYNTAX

The freeform syntax for listing database structure files with DMSLST appears in Figure 4-2. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSLST screen. For further discussion of the options, refer to the "Listing Database Files - Batch Mode" portion of this section. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

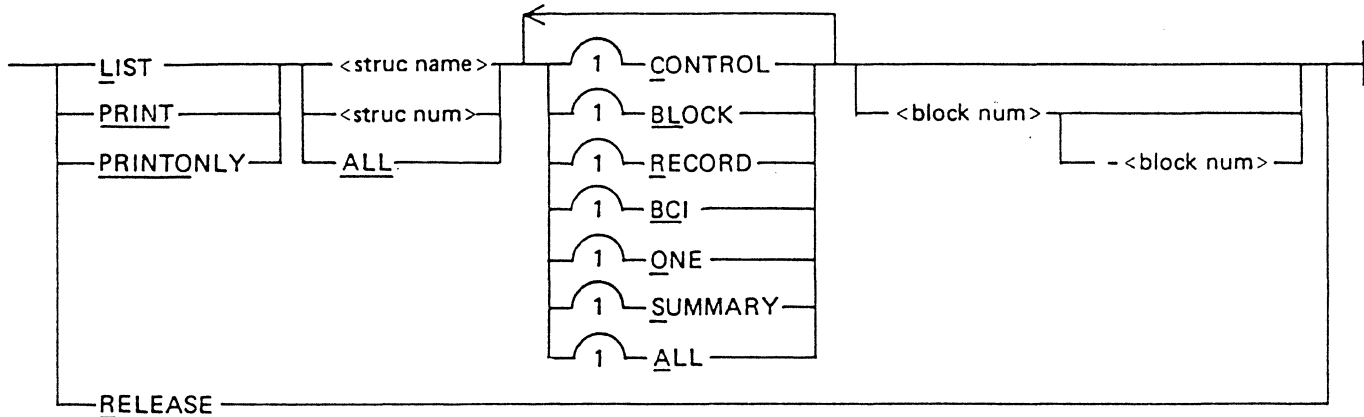


Figure 4-2. DMSUTL Freeform Syntax for Listing Database Files with DMSLST

LISTING DATABASE FILES - BATCH MODE

For general directions on the use of batch mode, see the "Using DMSUTL in Batch Mode" portion of Section 1. Figure 1-3 shows the general syntax; Figures 4-3 and 4-4 show the DMSLST specification syntax.

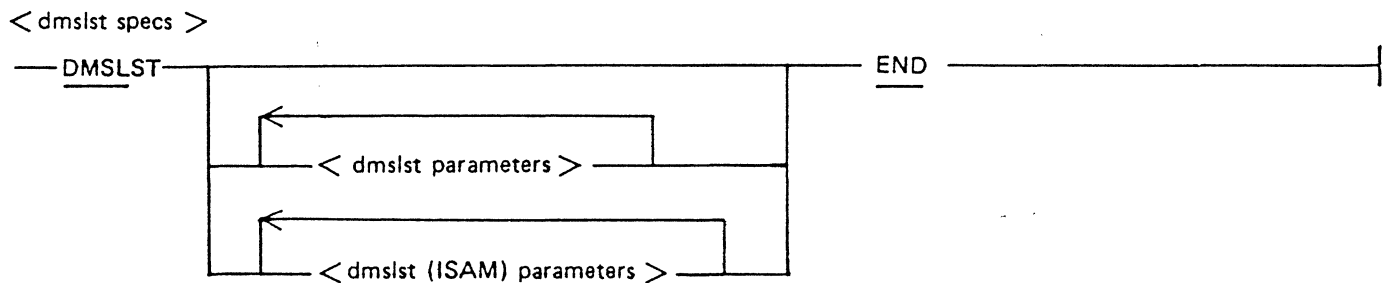


Figure 4-3. DMSLST Specification for Batch Use

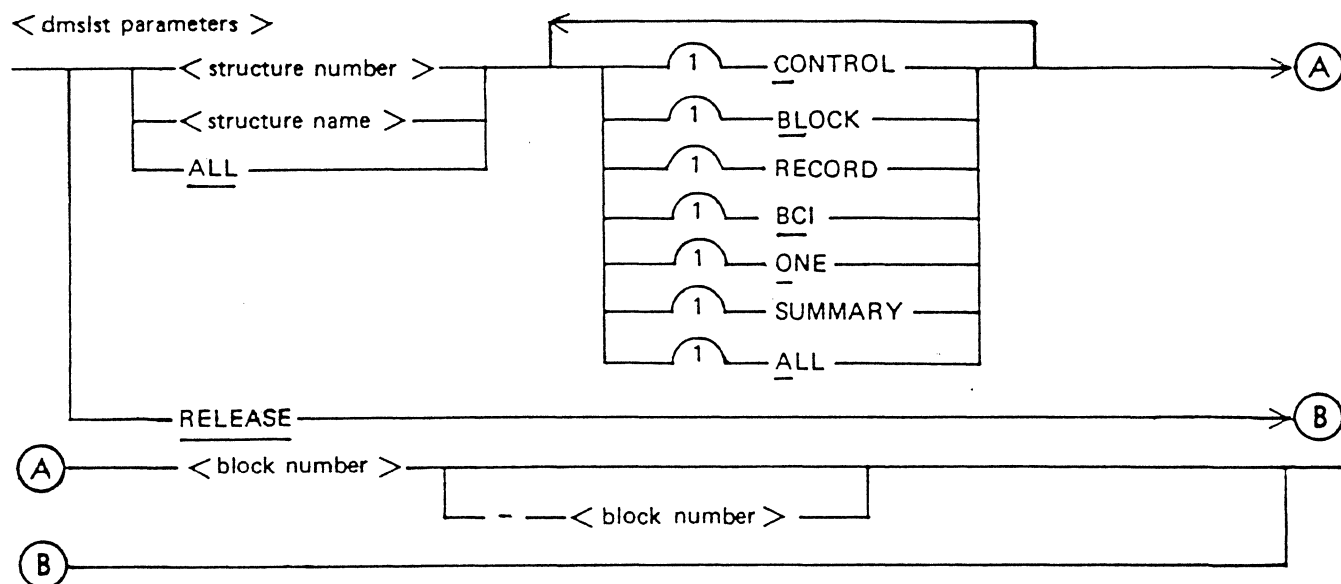


Figure 4-4. DMSLST Parameters for Listing Database Files - Batch Mode

RELEASE

Closes the current printer backup file so that you can print it.

<structure number>

The structure number has a maximum length of 4 digits.

<structure name>

The structure name has a maximum length of 17 characters.

ALL

With this parameter, the DMSLST program includes every structure, from 3 to the highest structure number.

CONTROL

With this parameter, the DMSLST program prints the contents of the record in the database control file (XXXCNF) corresponding to each selected structure.

BLOCK

With this parameter, the DMSLST program prints the entire contents of each selected block.

RECORD

With this parameter, the DMSLST program prints all the records in each selected block.

BCI

With this parameter, the DMSLST program prints the BCI for each selected block.

ONE

With this parameter, the DMSLST program prints Block one for each selected structure.

SUMMARY

With this parameter, the DMSLST program prints a brief summary of each selected structure.

ALL

With this parameter, the DMSLST program prints all of the above reports.

<block num>

The block number has a maximum length of 8 digits. The default value for block range is 2 - End Of File (EOF).

LISTING ISAM FILES - SCREEN MODE

Use the following procedure to list non-DMS ISAM files. Do not use this procedure for database structure files; use the procedure in the "Listing Database Files - Screen Mode" portion of this section.

On the home menu screen, enter the name of the ISAM file in the DATABASE DESIRED field and check the ISAM field. Then, press the TRANSMIT key. Figure 4-5 shows the screen that will appear on your terminal.

```

DMSLST ISAM MENU                                10:23 A.M.      DMSUTL
ACTION:  [                                     ]
          LIST PRInt print-ONLY RELease-printer
          REFresh HOME QUIT                      Press SPCFY for Help
-----
REPORT OPTIONS:

BLOCK      [ ]
RECORD     [ ]
BCI        [ ]
BLOCK ONE  [ ]

SPECIFIC FILE NAME OR KEY NUMBER: [           ]

BLOCK RANGE:  [           ] THROUGH [           ]
-----
```

Figure 4-5. DMSLST ISAM Menu Screen

1. REFresh

HOME

QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. LISt: The DMSUTL program displays the output on the terminal only.

3. PRInt: The DMSUTL program displays the output on the terminal and writes it to a printer backup file.

4. print-ONLY: The DMSUTL program writes the information to a printer backup file, but does not display it on the terminal.

5. RELease: The DMSUTL program closes the printer backup file so that you can print the output.

In the ACTION field, enter the capitalized letters to indicate your choice of action. The default value is LISt. Then, check the field opposite any of the following report options:

6. BLOCK: The DMSLST ISAM program will list/print the entire contents of the block(s) that you specify. Use the block range fields to select specific block ranges. If you do not specify a block range, the DMSLST ISAM program lists/prints all blocks. A sample copy of this report appears in Figure 4-9.

7. RECORD: The DMSLST ISAM program lists/prints the contents of the block(s) you specify and formats them into records. You can select a block range, if desired. If you do not specify a block range, the DMSLST ISAM program lists/prints all blocks. A sample copy of this report appears in Figure 4-10.

8. BCI: The DMSLST ISAM program will list/print Block Control Information (BCI) for the block(s) you specify. You can select a block range, if desired. If you do not specify a block range, the DMSLST ISAM program lists/prints all blocks. A sample copy of this report appears in Figure 4-11.

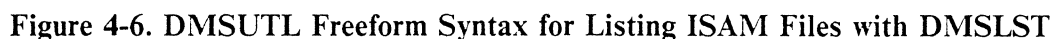
9. BLOCK ONE: The DMSLST ISAM program lists/prints the first block (file header) for the ISAM data or key file. A sample copy of this report appears in Figure 4-12.

10. SPECIFIC FILE NAME OR KEY NUMBER: To list/print the ISAM data file, enter either the file name or the number "0". To list/print a key file, enter either the key file name or its key number (1 through 99).

For example, if you had entered "FILEID" as the database name on the home menu screen, you would enter FILEID or 0 to indicate the ISAM data file. Key file names are XXXX01 through XXXX99 where XXXX represents the first four characters of the ISAM data file name. Therefore, for the same example, you could enter FILE01 or 1 to show the first key file.

11. BLOCK RANGE: Use these fields to select specific ranges of blocks. To see all blocks in the structure, leave both fields blank. To see only one block, enter a block number (1 to 8 digits) in the first field. To see a range of blocks, fill in both fields.

The freeform syntax for listing ISAM files with DMSLST appears in Figure 4-6. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSLST screen. For further discussion of the options, refer to the “Listing ISAM Files - Batch Mode” portion of this section. For a general description of the use of freeform syntax, see the “Home Menu” and “Using the DMSUTL Help Screens” portions of Section 1.



Use the DMSLST utility to list non-DMS ISAM files, in batch mode. For general directions on use of the batch mode, see the “Using DMSUTL in Batch Mode” portion of Section 1. Figure 1-3 shows the general syntax; Figures 4-3 and 4-7 show the DMSLST specification syntax to use for listing ISAM files.



B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

<file name>

File name is the 6-character identifier of an ISAM data file or key file.

<key number>

The key number is a 2-digit number referring to an ISAM data file or key file. You can use 0 to refer to the data file or the last 2 characters (numeric characters) of a key file's name to refer to that key file.

BLOCK

With this parameter, the DMSLST ISAM program prints the entire contents of each selected block.

RECORD

With this parameter, the DMSLST ISAM program prints all of the records in each selected block.

BCI

With this parameter, the DMSLST ISAM program prints the BCI for each selected block.

ONE

With this parameter, the DMSLST ISAM program prints Block one for each selected ISAM file (data file and key files).

ALL

With this parameter, the DMSLST ISAM program prints all of the above reports.

<block num>

The block number has a maximum length of 8 digits. The default value for block range is 2 - EOF.

RELEASE

Closes the current printer backup file so that you can print it.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

SAMPLE REPORTS

UNISYS B2000/B3000/B4000/V SERIES DMSII DATABASE LISTING REPORT FOR ADMIN ON THINGS

DATE: 09/30/87 TIME: 04:25 P.M. Database name
Time DMSLST program was run Location of DDF

THIS REPORT WAS GENERATED FOR THE FOLLOWING STRUCTURE(S): 13

THE FOLLOWING OPTIONS WERE SELECTED: CONTROL BLOCK RECORD BCI BLOCK ONE SUMMARY } Selected by user

THE BEGINNING BLOCK RANGE VALUE IS: 2
THE ENDING BLOCK RANGE VALUE IS: 2

CONTROL REPORT FOR STRUCTURE NUMBER 0013

ID : JOB-DESCRIPTIONS	FATHER STRUC NUM : 0000
VERSION : 872660052312806	BROTHER STRUC NUM : 0000
DISCONTINUITY TS : 872730057266331	BCI OFFSET : 02016
LAST UPDATE 1 : 872730058175275	BLOCKSIZE RECORDS : 0021
LAST UPDATE 2 : 872730058175275	STRUC CHECKSUM : TRUE
STATE : CLOSED	MODULUS : 00000000
DISCONTINUITY REQ : FALSE	FAMILYNAME : THINGS KIND : PACK
SYSTEM BUFFERS : 02	FIBDTK : 00 SINGLE PACK : FALSE
RANDOM USER BUFFERS : 01	FILLERS ZERO : TRUE
SERIAL USER BUFFERS : 02	WRITE AHEAD : FALSE
STRUC NUM : 0013	SOD TIME : 0000000000000000
STRUC TYPE : DATA SET	RECON STATUS : 0
ORGANIZATION : DIRECT	RECOVERY STATUS : NO RECOVERY
STRUC LEVEL : 00	BCI: ADDR CHECK WORD : 0000000000014
REC SIZE : 00096	CHECK SUM WORD : 36C716B10D67
DATA SIZE : 00072	ACN BLOCK : 0000000000000
CONTROL SIZE : 00000	ACN OFFSET : 000000
AREAS : 050	CHECKSUM : TRUE
AREALENGTH : 000021000	FORMAT LEVEL : 03
BLOCKSIZE : 02100	FORMAT : CONTROL
FAMILYNAME : THINGS	TYPE : DATA
DUPLICATES : NO DUPLICATES	RESERVED 1 : 000000000000

Figure 4-8. Sample Copy of DMSLST Control Report

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

UNISYS B2000/B3000/B4000/V SERIES DMSII DATABASE LISTING REPORT FOR ADMIN ON THINGS

DATE: 09/30/87 TIME: 04:25 P.M. ← Time DMSLST program was run

THIS REPORT WAS GENERATED FOR THE FOLLOWING STRUCTURE(S): 13

THE FOLLOWING OPTIONS WERE SELECTED: CONTROL BLOCK RECORD BCI BLOCK ONE SUMMARY } Selected by user

THE BEGINNING BLOCK RANGE VALUE IS: 2

THE ENDING BLOCK RANGE VALUE IS: 2

BLOCK REPORT FOR STRUCTURE NUMBER 0013

BLOCK NUM : 00000002

0000000100	0003E2E4D7	C5D9E5C9E2	D6D94040C6	0566086210	3910390000	(00060)
000000007F	4000000000	0000000000	0000000000	0002000002	D6D7E240D4	(00120)
C1D5C1C7C5	D940C60774	1631000000	0029900000	0100BF4000	0000000000	(00180)
0000000000	0000000003	000004C4C1	E3C140C3D3	C5D9D24040	E302492366	(00240)
0744111600	0000000000	7F23000000	0000000000	0000000000	0000040000	(00300)
01C4D740C4	C9D9C5C3E3	D6D940C610	0000000000	0000412500	001000BF40	(00360)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00420)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00480)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00540)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00600)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00660)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00720)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00780)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00840)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00900)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(00960)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01020)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01080)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01140)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01200)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01260)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01320)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01380)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01440)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01500)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01560)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01620)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01680)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01740)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01800)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01860)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01920)
0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	(01980)
0000000000	0000000000	0000000000	0000000013	00000002F3	9AC62203E2	(02040)
0000000001	5900013910	3020300000	0000001111	0000000000	0000000000	

Digit offset within block; count for digit at end of line

Figure 4-9. Sample Copy of DMSLST Block Report

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

UNISYS B2000/B3000/B4000/V SERIES DMSII DATABASE LISTING REPORT FOR ADMIN ON THINGS

DATE: 09/30/87 TIME: 04:25 P.M. Database name
Time DMSLST program was run

THIS REPORT WAS GENERATED FOR THE FOLLOWING STRUCTURE(S): 13

THE FOLLOWING OPTIONS WERE SELECTED: CONTROL BLOCK RECORD BCI BLOCK ONE SUMMARY

THE BEGINNING BLOCK RANGE VALUE IS: 2
THE ENDING BLOCK RANGE VALUE IS: 2

Block number RECORD REPORT FOR STRUCTURE NUMBER 0013
"Slot" within block of this record

BLOCK NUM : 00000002 SLOT NUM : 0001 PS : PRESENT
{ 0000000100 0003E2E4D7 C5D9E5C9E2 D6D94040C6 0566086210 3910390000 (00060)
000000007F 4000000000 0000000000 000000
Contents of record Offset within record: counts for digit at
end of line on which it appears

BLOCK NUM : 00000002 SLOT NUM : 0002 PS : PRESENT
0000000200 0002D6D7E2 40D4C1D5C1 C7C5D940C6 0774163100 0000002990 (00060)
000001008F 4000000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0003 PS : PRESENT
0000000300 0004C4C1E3 C140C3D3C5 D9D24040E3 0249236607 4411160000 (00060)
000000007F 2300000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0004 PS : PRESENT
0000000400 0001C4D740 C4C9D9C5C3 E3D6D940C6 1000000000 0000004125 (00060)
000010008F 4000000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0005 PS : NOT PRESENT
0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 (00060)
0000000000 0000000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0006 PS : NOT PRESENT
0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 (00060)
0000000000 0000000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0007 PS : NOT PRESENT
0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 (00060)
0000000000 0000000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0008 PS : NOT PRESENT
0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 (00060)
0000000000 0000000000 0000000000 000000

BLOCK NUM : 00000002 SLOT NUM : 0009 PS : NOT PRESENT
0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 (00060)
0000000000 0000000000 0000000000 000000

•
•
•

Figure 4-10. Sample Copy of DMSLST Record Report

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

UNISYS B2000/B3000/B4000/V SERIES DMSII DATABASE LISTING REPORT FOR ADMIN ON THINGS

Date
DMSLST was run → DATE: 09/30/87 TIME: 04:25 P.M.

THIS REPORT WAS GENERATED FOR THE FOLLOWING STRUCTURE(S): 13

THE FOLLOWING OPTIONS WERE SELECTED: CONTROL BLOCK RECORD BCI BLOCK ONE SUMMARY

THE BEGINNING BLOCK RANGE VALUE IS: 2
THE ENDING BLOCK RANGE VALUE IS: 2

} Selected by user

BCI REPORT FOR STRUCTURE NUMBER 0013

BCI FOR BLOCK NUM : 00000002

ADDR CHECK
STRUC NUM : 0013
BLOCK NUM : 00000002
CHECK SUM : F39AC62203E2
AUDIT CHANGE NUM
BLOCK NUM : 000000000159 ← Audit block number containing record of last update to this block
REC OFFSET : 000139 ← Offset of record written audit block
CHECKSUM : TRUE
FORMAT LEVEL : 03 ← Format level of BCI
FORMAT : DIRECT
TYPE : DATA
PRESENCE DIGITS :
11110000000000000000

Figure 4-11. Sample Copy of DMSLST BCI Report

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

```
UNISYS B2000/B3000/B4000/V SERIES DMSII DATABASE LISTING REPORT FOR ADMIN ON THINGS
DATE: 09/30/87  TIME: 04:25 P.M.
THIS REPORT WAS GENERATED FOR THE FOLLOWING STRUCTURE(S): 13
THE FOLLOWING OPTIONS WERE SELECTED: CONTROL BLOCK RECORD BCI BLOCK ONE SUMMARY
THE BEGINNING BLOCK RANGE VALUE IS:      2
THE ENDING BLOCK RANGE VALUE IS:        2
BLOCK ONE REPORT FOR STRUCTURE NUMBER 0013

BLOCK ONE INFO

LAST UPDATE
  DATE : 87273
  TIME : 0058175275 } Date-timestamp format
EOF BLOCK : 00000002 ← Number of last block in file
FREE BLOCK : 00000002 ← Number of 1st free block
FORMAT LEVEL : 03 ← Unisys use
SET ROOT/ SET HEAD : 00000000 ← Block number where root table starts
SET FIRST FINE/ SET TAIL : 00000000 ← Block number where fine table starts
SET LAST FINE : 00000000 ← Block number of end of fine table
SET NO LEVELS : 00 ← Number of levels of indexing (e.g., 2 coarse, 1 fine - no. levels = 5)
DUMP DATE TIME STAMP : 0000000000000000
SOD AUDIT FILE NAME :
SOD AUDIT BLOCK NUMBER : 00000000000000
SOD AUDIT BLOCK OFFSET : 000000
```

Figure 4-12. Sample Copy of DMSLST Block One Report

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Structure Listing Utility (DMSLST)

UNISYS B2000/B3000/B4000/V SERIES DMSII DATABASE LISTING REPORT FOR ADMIN ON THINGS

DATE: 09/30/87 TIME: 04:25 P.M. Database name Location of DDF

Time DMSLST program was run

THIS REPORT WAS GENERATED FOR THE FOLLOWING STRUCTURE(S): 13

THE FOLLOWING OPTIONS WERE SELECTED: CONTROL BLOCK RECORD BCI BLOCK ONE SUMMARY } Selected by user

THE BEGINNING BLOCK RANGE VALUE IS: 2

THE ENDING BLOCK RANGE VALUE IS: 2

SUMMARY REPORT

ADM013 JOB-DESCRIPTIONS DIRECT DATA SET

ON THINGS DISJOINT VERSION : 09/23/87 @ 0014:31:52:806

Figure 4-13. Sample Copy of DMSLST Summary Report

SECTION 5

CONTROL FILE PRINT UTILITY (DMSCTL)

INTRODUCTION

The DMSCTL utility provides a report of the information in the database control file (XXXCNF). An example of the report appears in Figure 5-4.

INITIATING DMSCTL - SCREEN MODE

To use DMSCTL with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 3 in the YOUR CHOICE IS field on the home menu screen. Put the name of the database in the DATABASE DESIRED field; put the location (DISK, PACK, or <diskpack name>) of the control file and Data Definition File (DDF) in the ON field. Then, press the TRANSMIT key.

Figure 5-1 shows the screen that will appear on your terminal.

ACTION:	DMSCTL MENU [REFresh HOME QUIT]	10:23 A.M.	DMSUTL
		Press SPCFY for Help	
COMMAND OPTIONS:			
1. LIST			
2. PRINT			
3. PRINTONLY			
4. RELEASE PRINTER			
YOUR CHOICE IS: []			

Figure 5-1. DMSCTL Menu Screen

1. REFresh
HOMe
QUIT

For an explanation of these selections, see the “Using the DMSUTL Data Entry Screens” portion of Section 1.

2. LIST: The DMSUTL program displays the output on the terminal only.
3. PRINT: The DMSUTL program displays the output on the terminal and writes it to a printer backup file.
4. PRINTONLY: The DMSUTL program writes the information to a printer backup file, but does not display it on the terminal.
5. RELEASE PRINTER: The DMSUTL program closes the printer backup file so that you can print the output.
6. YOUR CHOICE IS: Enter 1, 2, 3, or 4.

FREEFORM SYNTAX

The freeform syntax for DMSCTL appears in Figure 5-2. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSCTL screen. For further discussion of the options, refer to the “Initiating DMSCTL - Batch Mode” portion of this section. For a general description of the use of freeform syntax, refer to the “Home Menu” and “Using the DMSUTL Help Screens” portions of Section 1.

<u>L</u> IST
<u>P</u> RI <u>N</u> T
<u>P</u> RI <u>N</u> T <u>O</u> N <u>L</u> Y
<u>R</u> EL <u>E</u> A <u>S</u> E

Figure 5-2. DMSUTL Freeform Syntax for DMSCTL

INITIATING DMSCTL - BATCH MODE

For general directions on use of the batch mode, refer to the "Using DMSUTL in Batch Mode" portion of Section 1. Figure 1-3 shows the general syntax; Figure 5-3 shows the DMSCTL specification syntax.

< dmsctl specs >

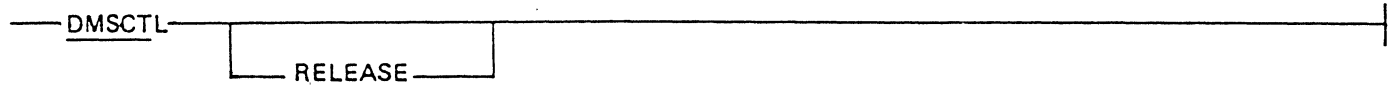


Figure 5-3. DMSCTL Specification for Batch Use

RELEASE

Closes the current printer backup file so that you can print it.

DMSCTL REPORT

A sample of part of a DMSCTL report, which shows a header record report and a structure report for a data set, appears in Figure 5-4. Following the figure are definitions of all terms used in the report. Refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration* for more information on the control file.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Control File Print Utility (DMSCTL)

UNISYS B2000/B3000/B4000/V SERIES DMSII CONTROL FILE REPORT FOR DATA BASE: ADMIN ON THINGS
DATE: 09/30/87 TIME: 04:15 P.M.

HEADER RECORD	
IN USE : NO RECOVERY	BCI
HIGHEST STRUC NUM : 20	ADDR CHECK WORD : 000000000001
VERSION : 872730038540690	CHECK SUM WORD : 6D667DE87406
CREATION : 872660052312806	ACN BLOCK : 0
RELEASE LEVEL : 7200	ACN OFFSET : 0
FORMAT LEVEL : 3	CHECKSUM : TRUE
STATISTICS : TRUE	FORMAT LEVEL : 3
GLOBAL DATA : FALSE	FORMAT : CONTROL
AUDIT PARAMS	TYPE : DATA
AUDITED : SINGLE AUDIT	RESERVED 1 : 000000000000
SYNC : 5	
CONT : 2	
AUDIT CHECKSUM : TRUE	
AUDIT FILE NAME : ADMA04	
NEXT AUDIT BLOCK : 160	
UPDATE EOF : 100	
CONTROL FILE LOCK ID : NOT LOCKED	
CONTROL FILE LOCK FUNCTION : 00	
DBP ID : ADMIN	
RESULT	
CAT : SUCCESSFUL	
SUB CAT : 0	
STRUC : 0	
OCCUR : 0	
ALLOWED CORE : 750000	
RECOVERY VERIFICATION : NO RECOVERY	
SOD TIME : 0000000000000000	
SOD FILE :	
SOD BLOCK : 000000000000	
SOD OFFSET : 000000	
REC TIME : 0000000000000000	
RECON STAT : 0	
RECON FILE :	
RECON BLOCK : 000000000000	
RECON OFFSET : 000000	
LOCKED OUT CNT : 0	
REQ-DATE-TIME-S: 0000000000000000	

Figure 5-4. Sample of DMSCTL Report (Sheet 1 of 2)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Control File Print Utility (DMSCTL)

CONTROL FILE STRUCTURE RECORDS

CONTROL FILE RRN 2	CONTROL FILE RRN 3	CONTROL FILE RRN 4
ID : STATISTICS	ID : GLOBAL DATA	ID : RESTART-RECORD
VERSION : 872730038540690	VERSION : 872660052312806	VERSION : 872660052312806
DISCONTINUITY TS : 872730038540690	DISCONTINUITY TS : 872660052312806	DISCONTINUITY TS : 872730057281673
LAST UPDATE 1 : 872730058162494	LAST UPDATE 1 : 0000000000000000	LAST UPDATE 1 : 872730058176844
LAST UPDATE 2 : 872730058162494	LAST UPDATE 2 : 0000000000000000	LAST UPDATE 2 : 872730058176844
STATE : CLOSED	STATE : UNASSIGNED	STATE : CLOSED
DISCONTINUITY REQ : FALSE	DISCONTINUITY REQ : FALSE	DISCONTINUITY REQ : FALSE
SYSTEM BUFFERS : 0	SYSTEM BUFFERS : 0	SYSTEM BUFFERS : 1
RANDOM USER BUFFERS : 0	RANDOM USER BUFFERS : 0	RANDOM USER BUFFERS : 0
SERIAL USER BUFFERS : 0	SERIAL USER BUFFERS : 0	SERIAL USER BUFFERS : 0
STRUC NUM : 1	STRUC NUM : 2	STRUC NUM : 3
STRUC TYPE : STATS DATA	STRUC TYPE : GLOBAL DATA	STRUC TYPE : RESTART DS
ORGANIZATION : STANDARD	ORGANIZATION : DIRECT	ORGANIZATION : STANDARD
STRUC LEVEL : 0	STRUC LEVEL : 0	STRUC LEVEL : 0
REC SIZE : 300	REC SIZE : 120	REC SIZE : 108
DATA SIZE : 300	DATA SIZE : 0	DATA SIZE : 108
CONTROL SIZE : 0	CONTROL SIZE : 0	CONTROL SIZE : 0
AREAS : 99	AREAS : 1	AREAS : 1
AREALENGTH : 20	AREALENGTH : 120	AREALENGTH : 12272
BLOCKSIZE : 360	BLOCKSIZE : 100	BLOCKSIZE : 944
FAMILYNAME : THINGS	FAMILYNAME : THINGS	FAMILYNAME : THINGS
DUPLICATES : NO DUPLICATES	DUPLICATES : NO DUPLICATES	DUPLICATES : NO DUPLICATES
FATHER STRUC NUM : 0	FATHER STRUC NUM : 0	FATHER STRUC NUM : 0
BROTHER STRUC NUM : 0	BROTHER STRUC NUM : 0	BROTHER STRUC NUM : 0
BCI OFFSET : 300	BCI OFFSET : 200	BCI OFFSET : 864
BLOCKSIZE RECORDS : 1	BLOCKSIZE RECORDS : 1	BLOCKSIZE RECORDS : 8
STRUC CHECKSUM : TRUE	STRUC CHECKSUM : TRUE	STRUC CHECKSUM : TRUE
MODULUS : 0	MODULUS : 0	MODULUS : 0
FAMILYNAME : THINGS	FAMILYNAME : THINGS	FAMILYNAME : THINGS
KIND : PACK	KIND : PACK	KIND : PACK
FIBOTK VALUE : 0	FIBOTK VALUE : 0	FIBOTK VALUE : 0
SINGLE PACK : FALSE	SINGLE PACK : FALSE	SINGLE PACK : FALSE
FILLERS ZERO : FALSE	FILLERS ZERO : FALSE	FILLERS ZERO : TRUE
WRITE AHEAD : FALSE	WRITE AHEAD : TRUE	WRITE AHEAD : TRUE
SOD TIME : 0000000000000000	SOD TIME : 0000000000000000	SOD TIME : 0000000000000000
RECON STAT : 0	RECON STAT : 0	RECON STAT : 0
RECOVERY STATUS : NO RECOVERY	RECOVERY STATUS : NO RECOVERY	RECOVERY STATUS : NO RECOVERY
DATASET- REF : 0	DATASET- REF : 0	DATASET- REF : 0
BCI	BCI	BCI
ADDR CHECK WORD : 000000000002	ADDR CHECK WORD : 000000000003	ADDR CHECK WORD : 000000000004
CHECK SUM WORD : ECA8D3A23DDC	CHECK SUM WORD : 8A691D4D3DE0	CHECK SUM WORD : 52160C8FA53A
ACN BLOCK : 0	ACN BLOCK : 0	ACN BLOCK : 0
ACN OFFSET : 0	ACN OFFSET : 0	ACN OFFSET : 0
CHECKSUM : TRUE	CHECKSUM : TRUE	CHECKSUM : TRUE
FORMAT LEVEL : 3	FORMAT LEVEL : 3	FORMAT LEVEL : 3
FORMAT : CONTROL	FORMAT : CONTROL	FORMAT : CONTROL
TYPE : DATA	TYPE : DATA	TYPE : DATA
RESERVED 1 : 000000000000	RESERVED 1 : 000000000000	RESERVED 1 : 000000000000

Figure 5-4. Sample of DMSCTL Report (Sheet 2 of 2)

Heading

The heading consists of the database name and the location of the control file. The database name follows the colon (:). The location of the control file (DISK, PACK, or <diskpack name>) follows "ON." PACK indicates any unreserved diskpack.

Date and Time

The date and time that DMSCTL was run appear under the heading. This date and time represent the time that you initiated DMSCTL; it does not come from the control file.

The first page of the report describes the content of the control file HEADER RECORD (Block one). The report lists the values contained in the following fields:

IN USE

This is the status value contained in the in-use flag for this database.

HIGHEST STRUC NUM

This is the highest structure number that exists in this database.

VERSION

This is the date-timestamp of the last DASDL UPDATE compile or REORGANIZE that was done against this database.

CREATION

This is the date-timestamp of the original DASDL compile that created this database.

RELEASE LEVEL

This is the release of DMSII that was used to create this control file. For example, 7100 indicates that the 7.1 release was used.

FORMAT LEVEL

This information is for Unisys technical use only.

STATISTICS

This indicates whether or not statistics are being kept for this database. You set the value of this option in DASDL.

GLOBAL DATA

This is always FALSE.

AUDIT PARAMS

These are the values of the audit parameters you set in the DASDL audit trail specification for this database.

CONTROL FILE LOCK ID and CONTROL FILE LOCK FUNCTION

Values for these fields appear in the control file description in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

DBP ID

This is the name (6 characters) of the DBP code file (and of this database in general).

RESULT

This is the result of the last database operation done on this database. Values for all categories besides SUCCESSFUL appear in the table of DMSTATUS error conditions in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

ALLOWDCORE

This is the maximum size, in digits, that the DBP can expand to. It is the latest value you set in the DASDL source or with the SP command.

RECOVERY VERIFICATION

This gives the current status of recovery procedures.

SOD TIME

SOD FILE, BLOCK, and OFFSET

These fields will be empty unless this control file was loaded from an online dump tape and has not yet been recovered successfully with a rebuild or reconstruct recovery. In this case, these fields contain the date-timestamp and the audit pointer to the corresponding start of an online dump (SOD) record in the audit trail.

REC TIME

This is the date-timestamp of the last rebuild or reconstruct recovery that was done against this database.

RECON STAT

This is the status of online reconstruct. It can have the following values:

- 0 - No online reconstruct in progress (normal).
- 4 - Online reconstruct in progress.
- 8 - Online reconstruct aborted.

RECON FILE

RECON BLOCK

RECON OFFSET

These fields comprise a pointer to the last End of Online Reconstruct audit record in the audit trail for this database.

LOCKED OUT CNT

This is the number of structures in this database that are locked out.

REO-DATE-TIME-S:

This is the date-timestamp of the last reorganize that was done on this database.

BCI

The Block Control Information (BCI) on the report includes the following headings: ADDR CHECK, CHECKSUM, ACN BLOCK, ACN OFFSET, CHECKSUM Flag, FORMAT LEVEL, FORMAT, TYPE, and RESERVED 1.

BCI values appear in the BCI description in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*. ACN BLOCK and OFFSET are the block number and offset within that block of the audit file record concerning the last change to this block of the control file.

The report describes the contents of records 2 through n of the control file. Each record contains information corresponding to a specific structure file in the database. For each, the report lists the values contained in the following fields. Note that all date-timestamps in the report are in the following format: *yyddmmmmmmmmmmmm* where *yy* is the year, *ddd* is the Julian date, and *mmmmmmmmmmmm* is the time in milliseconds past midnight.

CONTROL FILE RRN

This is the number of the block (Relative Record Number) in the control file that contains the information about this structure.

ID

This is the name that you assigned to this structure in the DASDL source.

VERSION

This is the date-timestamp of the last DASDL UPDATE compile or REORGANIZATION that changed this structure.

DISCONTINUITY TS

This is the date-timestamp of the last structure file discontinuity in the audit trail pertaining to this structure. Rollback recovery cannot cross this point.

LAST UPDATE 1 and 2

These are the date-timestamps the DBP uses for recovery in case the operating system fails while this structure is being updated. By comparing these date-timestamps with the one in the structure file, the recovery routine can determine whether or not the update levels of the control file and the structure file match. If they are the same, the update levels match.

STATE

This is the state of this structure: open, closed, etc.

DISCONTINUITY REQ

If TRUE, a reorganization has been run on the database and the corresponding discontinuity record has not yet been written into the audit trail for this structure.

SYSTEM BUFFERS

RANDOM USER BUFFERS

SERIAL USER BUFFERS

These are values you set with the DASDL physical option specification or the SP command to control the number of buffers that are allocated for this structure.

STRUC NUMBER

The number assigned by the DASDL compiler to represent this structure. This number forms the last three characters of the structure's physical file name.

STRUC TYPE, ORGANIZATION

The values in these fields indicate the type and organization of this structure. You designated them in the DASDL source.

STRUC LEVEL

This indicates the embedded level of this structure. A disjoint structure is at level zero. A structure embedded in a disjoint structure is at level 1, and so on.

REC SIZE

This is the size of the records in this structure.

DATA SIZE

For data sets, this field contains the same value as REC SIZE. For sets and subsets, it contains the key size (entry size minus pointer size).

CONTROL SIZE

The control size is not used.

AREAS

This is the number of disk or diskpack areas this structure file currently takes up.

AREALENGTH, BLOCKSIZE, FAMILYNAME

These are the physical attributes and location of this structure. You designated them in the physical option specification in the DASDL source.

DUPLICATES

If you specify DUPLICATES, you permit duplicate entries in an index file. You designate this in the DASDL key structure specification.

FATHER STRUC NUM

This is the number of the structure in which this structure is embedded (zero indicates it is not embedded).

BROTHER STRUC NUM

This is the structure number of other structures embedded at the same level as this structure in the father structure.

BCI OFFSET

This is the offset of the Block Control Information in this structure's blocks.

BLOCKSIZE RECORDS

This is the blocking factor for this structure.

STRUC CHECKSUM

This indicates whether or not checksums are maintained for the blocks in this structure.

MODULUS

This is the number of blocks assigned to the base area of a random data set. You designate this in the DASDL source with the set or the access specification.

FAMILYNAME, KIND

This is the location of this structure. You designate this in the physical option specification in the DASDL source.

FIBDTK VALUE

Disk or pack assignment technique (refer to the *V Series MCP/VS Programming Reference Manual*).

SINGLE PACK

This indicates whether or not this structure is assigned to a single pack (Boolean value).

FILLERS ZERO

If TRUE, an old release of the DASDL compiler created this structure and the DBP cannot search it optimally. You can remedy the problem with a set balancing (garbage collection) operation (see *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration* for more information about garbage collection).

WRITEAHEAD

This indicates whether or not writeahead has been specified for this structure. You designate this with the physical option specification in the DASDL source or with the SP command.

SOD TIME

This is the date-timestamp of the most recent online dump.

RECON STAT

This is the status of online reconstruct. It can have the following values:

- 0 - No online reconstruct in progress (normal).
- 4 - Online reconstruct in progress.
- 8 - Online reconstruct aborted.

RECOVERY STATUS

The recovery process uses this as scratchpad memory.

DATA SET-REF

For a set or subset, this is the structure number of the data set that this structure references. This field is not used for a data set.

BCI

The BCI on the report includes the following headings: ADDR CHECK, CHECKSUM, ACN BLOCK, ACN OFFSET, CHECKSUM Flag, FORMAT LEVEL, FORMAT, TYPE, and RESERVED 1.

BCI values appear in the BCI description in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*. ACN BLOCK and OFFSET are the block number and offset within that block of the audit file record concerning the last change to this block of the control file.

SECTION 6

DATABASE ANALYSIS UTILITY (DMSDAP)

INTRODUCTION

The Data Management System II (DMSII) database analysis utility is a special purpose utility for checking the internal consistency of key, pointer, and control information in the structure files of a given database. It can be applied usefully in three main situations:

1. When you want an occasional check on the internal consistency of key, pointer, and control information in database structure files.
2. When you suspect that erroneous logic in a user program has invalidated some of the information in a database - for example, deletion of a record to which a reference exists in a manual subset.
3. When you want to check the conversion of nonDMSII files to DMSII format to verify that the conversion has been performed correctly.

DMSDAP makes no attempt to correct any inconsistencies that may be detected.

The DMSDAP program creates a summary report in a printer backup file called DMSUTL/<database name>.

INITIATING DMSDAP - SCREEN MODE

To use DMSDAP with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 5 in the YOUR CHOICE IS field on the home menu screen. Put the name of the database in the DATABASE DESIRED field; put the location (DISK, PACK, or <diskpack name>) of the database in the ON field. Then, press the TRANSMIT key.

Figure 6-1 shows the screen that will appear on your terminal.


```
ACTION:      DMSDAP MENU                      10:23 A.M.      DMSUTL
              [                               ]
              REFresh  HOME  QUIT
```

```
ALL STRUCTURES:  [ ]          OR
```

```
ONE STRUCTURE:   [          ]
```

```
EXTENDED VERIFY: [ ]
```

```
THE STRUCTURE NAME CAN BE EITHER THE STRUCTURE NUMBER
(3 TO 999) OR THE STRUCTURE NAME (AS DEFINED IN DASDL.)
```

```
NOTE: DMSDAP WILL RUN IN BATCH MODE IN EITHER CASE.
```

Figure 6-1. DMSDAP Menu Screen

1. REFresh
HOME
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. ALL STRUCTURES: When you check this field (with any nonblank character), the DMSDAP program analyzes all of the structures in the database.
3. ONE STRUCTURE: Enter either a structure name, as defined in DASDL, or a structure number from 3 through 999.
4. EXTENDED VERIFY: If you do not check this field, the DMSDAP program checks Block Control Information (BCI) and the ordering of index keys. If you do check this field, the DMSDAP program also checks the pointers and index entries against structure files. An extended verify takes longer than a regular verify. For details, see the "Description of DMSDAP Analysis" portion of this section.

When you select item 2 or 3, DMSDAP executes as a batch job.

FREEFORM SYNTAX

The freeform syntax for DMSDAP appears in Figure 6-2. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSDAP screen. For further discussion of the options, refer to the "Initiating DMSDAP - Batch Mode" portion of this section. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

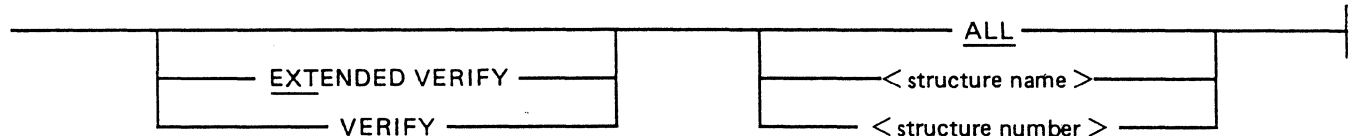


Figure 6-2. DMSUTL Freeform Syntax for DMSDAP

INITIATING DMSDAP - BATCH MODE

For general directions on use of the batch mode, refer to the "Using DMSUTL in Batch Mode" portion of Section 1. Figure 1-3 shows the general syntax; Figure 6-3 shows the DMSDAP specification syntax.

< dmsdap specs >

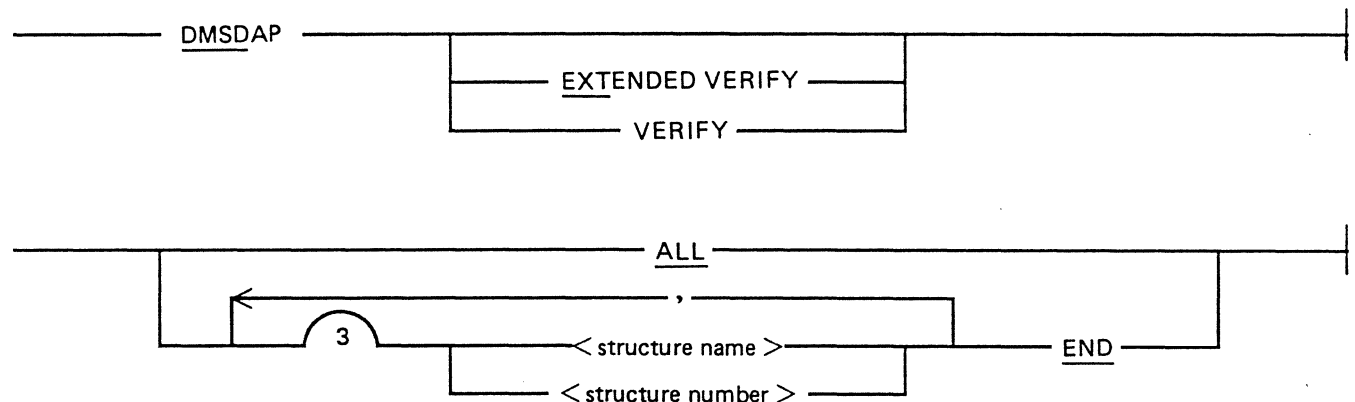


Figure 6-3. DMSDAP Specification for Batch Use

If you select neither VERIFY nor EXTENDED VERIFY, the default value is VERIFY.

<structure name>

The structure name is the name of the structure you want the DMSDAP program to analyze. The name should be the same as that defined in DASDL and cannot exceed 17 characters.

<structure number>

The structure number is the number (up to 3 digits) of the structure you want the DMSDAP program to analyze. You can obtain the structure number from the report produced by the DMSCTL utility.

ALL

When you specify ALL, you request the DMSDAP program to analyze all of the structures in the database.

DESCRIPTION OF DMSDAP ANALYSIS

The process of validating a structure consists of checking various control fields and the relationships between the data sets, sets, and subsets.

Regular verification is quicker than extended verification because it avoids the random I/Os associated with an extended verification.

If the selected structure is a data set, the DMSDAP program makes the following consistency checks:

- It checks the BCI in each non-null block with regard to: (a) structure number and block number (address check), (b) structure type, (c) block type, (d) agreement between the count of non-null records in block and the actual number of non-null records in the block (standard data set only), and correctness of CHECKSUM (if set).
- If you select the EXTENDED VERIFY option, it also checks the pointers for each set or subset embedded in this data set to ensure that: (a) the table pointed to by the first pointer is a root table or a root-fine table, (b) the table pointed to by the second pointer is a fine, head, or head-tail table with a null previous table pointer, (c) the table pointed to by the third pointer is a fine-tail or head-tail table with a null next table pointer, and (d) no table is pointed to more than once.

If the selected structure is a set or subset, the DMSDAP program makes the following consistency checks:

- It checks the BCI in each table (block) to validate: (a) structure number and block number (address check), (b) structure type, (c) table type, and (d) count of non-null entries against actual number of non-null entries.
- It verifies the order of keys in each table.
- If the CHECKSUM option is SET so that checksums are being calculated, it checks the checksums.
- If you select the EXTENDED VERIFY option, it also verifies the pointers and order of keys between tables on the same level.
- If you select the EXTENDED VERIFY option, it checks each fine table entry against the corresponding data set record to ensure that the keys are the same in the set and data set. In addition, it keeps track of the data set records so that it can verify that all data set records have a corresponding set entry, and that no data set records have multiple set entries.

DMSDAP REPORT

If there are inconsistencies in a DMS structure reported on, the DMSDAP report gives the error number of each inconsistency. A list of error numbers appears here with a brief explanation of each error message. Refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* for an explanation of valid field values.

0 STRUCTURE NAME DOES NOT COMMENCE WITH A LETTER

Structure name you requested is invalid because it does not begin with a letter.

1 STRUCTURE NAME CONTAINS ILLEGAL CHARACTERS

Structure name you requested is invalid because it contains illegal characters.

2 MORE THAN 17 CHARACTERS IN STRUCTURE NAME

Structure name you requested is invalid because it contains more than 17 characters.

3 STRUCTURE NAME ENDS IN HYPHEN

Structure name you requested is invalid because it ends in a hyphen.

4 NO STRUCTURE WITH THIS NAME

Structure name you requested is invalid because it is not the name of a structure in the database, as described in DASDL. Check to be sure you entered the name correctly.

5 INVALID STRUCTURE NUMBER IN DDF EXTRACT RECORD

The DMSDAP program reads the Data Definition File (DDF) to produce a disk file called the DDF extract file. A structure number in this file is invalid. This indicates a problem in either the DDF file or a database structure file.

6 MISMATCH BETWEEN COUNT AND PRESENCE DIGITS

The BCI is invalid for a block because the non-null record count does not agree with the number of records that was indicated by the presence digits.

7 NULL TABLE

This table contains no entries, but pointers to it still exist. These pointers can be from other tables or from data set records.

8 PREVIOUS POINTER SHOULD BE NULL IN THIS FINE TABLE

This table should be the first fine table in a string of fine tables, but it has a pointer to a previous fine table.

9 NEXT POINTER SHOULD BE NULL IN THIS FINE TABLE

This table should be the last fine table in a string of fine tables, but it has a pointer to a next fine table.

10 INVALID KEY TYPE

The key type indicated in the DDF is invalid. A problem exists with the DDF or with the DMSDAP analysis of the DDF.

11 KEY MISMATCH TYPE 2 OR 3

Keys are not in proper order. For TYPE 2, the keys out of sequence are in two different levels of the tables. For TYPE 3, the keys out of sequence are between two tables in the same level.

12 IDENTICAL KEYS BUT DUPLICATES NOT ALLOWED

The DMSDAP program has found keys that are identical, but duplicate keys are not allowed in the set.

13 TWO KEY ITEMS OUT OF ORDER IN TABLE

A table has keys out of sequence.

14 INVALID TABLE TYPE

This BCI is invalid for this table because it specifies a particular table type. The structure of the other tables implies that this table is another type.

15 TABLE PREVIOUS AND NEXT POINTER MISMATCH

Pointers between tables on the same level do not agree.

16 INVALID OR OUT OF RANGE POINTER VALUE

A pointer has a value that is not numeric or that points to a block that does not exist.

17 KEY MISMATCH TYPE 1

The value of the highest key in a table (Table X) does not match the key in the table entry in the table (Table Y) pointing to Table X. Table Y is either a coarse table or a root table; an entry in a coarse or root table points to another table whose highest key entry value should equal the key value of the entry in the coarse or root table.

18 SEVERAL POINTERS TO THIS TABLE

There are several tables with pointers to this table.

19 POINTER IN TABLE TO NULL RECORD

A pointer to a data set record that does not exist appears in a set or subset table.

20 SEVERAL POINTERS TO THIS RECORD

This data set record has several pointers to it. This condition may be valid if pointers are from an embedded subset.

21 ROOT TABLE EXPECTED

The DMSDAP program expects a root table, but the BCI shows a different table type.

22 FINE TABLE EXPECTED

The DMSDAP program expects a fine table, but the BCI shows another table type.

23 KEY MISMATCH BETWEEN KEY IN TABLE SLOT AND KEY IN DATA RECORD

The key in a fine table entry does not agree with the key stored in the data set record it points to.

- 24 SEVERAL POINTERS TO THIS TABLE FROM ABOVE
Several tables on the same level point to this table by means of their *next* pointers.
- 25 SEVERAL POINTERS TO THIS TABLE FROM BELOW
Several tables on the same level point to this table by means of their *previous* pointers.
- 26 INVALID OR OUT OF RANGE POINTER VALUE
A pointer has a value that is not numeric or that points to a block that does not exist.
- 27 NO POINTERS TO THIS TABLE
This table has valid entries, but no pointers point to it.
- 28 INVALID ADDRESS CHECK IN BLOCK CONTROL INFO
The BCI is invalid because the address check field is invalid. The address check consists of a 4-digit structure number and an 8-digit block number.
- 29 INVALID STRUCTURE TYPE IN BLOCK CONTROL INFO
The BCI is invalid because it contains a structure type that is not valid for the structure.
- 30 INVALID BLOCK TYPE IN BLOCK CONTROL INFO
The BCI is invalid because it contains a block type that is invalid.
- 31 INVALID CHECKSUM IN THE BLOCK READ
The computed checksum value does not match the value recorded in the block's BCI.
- 32 POINTER BEYOND END OF FILE
A set or subset pointer exists to a record beyond the logical End Of File (EOF) of the referenced data set.
- 33 HEAD POINTER POINTING BACK TO BLOCK OF INVALID TYPE
The block pointed to is not a head or head-tail block for an unordered data set, unordered list, or ordered list set or subset.
- 34 HEAD TAIL POINTER MISMATCH
The head and tail pointers of an embedded unordered data set are not properly linked.
- 35 PREV POINTER MISMATCH
Two adjacent set or subset tables are not properly linked.
- 36 TAIL BLOCK, TAIL POINTER CONFLICT IN SUBFILE
The block in an embedded unordered data set pointed to by the master data set tail pointer is not the correct tail block.
- 37 LAST FINE DOES NOT MATCH WITH BLOCK ONE
The last fine table of an indexed sequential set or subset is not the same block as recorded in Block one.

38 FORMAT MUST BE RANDOM

A block of a random data set has the wrong format type flag.

39 BLOCK TYPE MUST BE BASIC BLOCK

A modulus block of a random data set is not flagged as a BASIC block.

40 INVALID PREV POINTER IN THE BLOCK

A block of a random or embedded unordered data set is improperly linked.

41 INVALID BASIC POINTER IN BLOCK

An overflow block of a random data set is linked to the wrong basic block.

42 INVALID OR OUT OF RANGE RESIDENT POINTER

An embedded unordered list set or subset (that is resident in a master record) contains an invalid pointer to the referenced data set, or the resident count is invalid.

43 RESIDENT POINTER TO NULL RECORD

A resident pointer of an embedded unordered set or subset points to a record that is not present in the referenced data set.

DMSDAP also provides information on the physical attributes and physical mapping of the structures. The following list shows the information the DMSDAP program provides for the structures:

ANALYSIS OF STRUCTURE

This shows the structure number and the logical structure name of the structure.

STRUCTURE TYPE

This shows the organization of the structure (for example, standard data set).

EMBEDDED LEVEL

Embedded level refers to the number of levels the structure is embedded. All disjoint structures have an embedded level of zero; a structure embedded within a disjoint structure has a level of one, etc.

BCI OFFSET IN DIGITS

This represents the offset, in digits, to the Block Control Information.

NUMBER OF AREAS

This field shows the maximum number of areas the physical file can have.

AREA LENGTH IN BYTES

This field shows the length of each area in bytes.

AREA LENGTH IN BLOCKS

This field shows the length of each area in physical blocks.

TOTAL BLOCKS IN THE FILE

This field shows the total number of blocks in the file.

TOTAL BLOCK SIZE IN DIGITS

This field shows the physical block size in digits. This includes the space necessary for the Block Control Information.

WASTED SPACE PER BLOCK IN DIGITS

This field shows the space wasted by each physical block because each block starts on a sector (100- or 180-byte) boundary.

NUMBER OF RECORDS PER BLOCK

This field shows the number of logical data set records in a block.

NUMBER OF ENTRIES PER BLOCK

This field shows the number of key entries that can be in a physical block of a set or subset.

RECORD SIZE IN DIGITS

This field shows the size of a data set record in digits.

ENTRY SIZE IN DIGITS

This field shows the size of an entry in a table of a set or subset.

DATA SIZE IN DIGITS

This field shows the size of the data in a data set record. This does not include any storage necessary in the record for embedded sets or subsets.

KEY SIZE IN DIGITS

This field shows the size of the key used in the set or subset.

POPULATION IN RECORDS

This field shows the number of data set records that have been stored in the data set.

POPULATION IN TABLES

This field shows the number of tables that have been written for the set or subset.

NUMBER OF TABLE LEVELS

This field shows the number of table levels that exist in a disjoint set or subset. This does not appear for embedded sets or subsets.

SECTION 7

DDF PRINT UTILITY (DDFLIS)

INTRODUCTION

The DDFLIS program provides a report of the information in the Data Definition File (DDF). The DDFLIS program produces a printer backup file called DMSUTL/<database name>. Be aware that this is quite a sizable listing. Use this listing for sophisticated error analysis in cases where you suspect that the contents of the DDF are questionable. This listing is not required for normal database management functions.

INITIATING DDFLIS - SCREEN MODE

To use DDFLIS with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 6 in the YOUR CHOICE IS field on the home menu screen. Put the name of the database in the DATABASE DESIRED field; put the location (DISK, PACK, or <diskpack name>) of the control file and DDF in the ON field. Then, press the TRANSMIT key.

Figure 7-1 shows the screen that will appear on your terminal.

```
ACTION:      DDFLIS MENU                      10:23 A.M.      DMSUTL
             [                                ]
             REFresh  HOME  QUIT              Press SPCFY for Help

COMMAND OPTIONS:

1. LIST
2. PRINT
3. PRINTONLY
4. RELEASE PRINTER

YOUR CHOICE IS:  [ ]
```

Figure 7-1. DDFLIS Menu Screen

1. REFresh
HOMe
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. LIST: The DMSUTL program displays the output on the terminal only.
3. PRINT: The DMSUTL program displays the output on the terminal and writes it to a printer backup file.
4. PRINTONLY: The DMSUTL program writes the information to a printer backup file, but does not display it on the terminal.
5. RELEASE PRINTER: The DMSUTL program closes the printer backup file so that you can print the output. If you do not release the file, you can write additional information to it (from another utility) and then release it.
6. YOUR CHOICE IS: Enter 1, 2, 3, or 4.

FREEFORM SYNTAX

The freeform syntax for DDFLIS appears in Figure 7-2. You can enter this syntax on the bottom of the Home menu screen. The options correspond to the fields and choices on the DDFLIS screen. For further discussion of the options, refer to the "Initiating DDFLIS - Batch Mode" portion of this section. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

<u>L</u> IST
<u>P</u> RI <u>N</u> T
<u>P</u> RI <u>N</u> T <u>O</u> N <u>L</u> Y
<u>R</u> EL <u>E</u> A <u>S</u> E

Figure 7-2. DMSUTL Freeform Syntax for DDFLIS

INITIATING DDFLIS - BATCH MODE

For general directions on use of the batch mode, refer to the "Using DMSUTL in Batch Mode" portion of Section 1. Figure 1-3 shows the general syntax; Figure 7-3 shows the DDFLIS specification syntax.

< ddfliis specs >

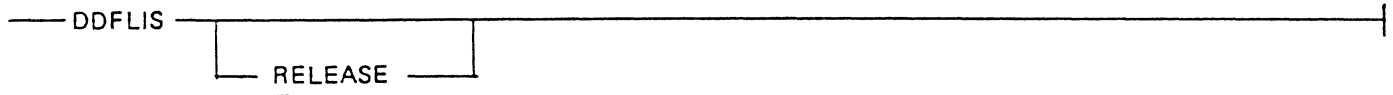


Figure 7-3. DDFLIS Specification for Batch Use

RELEASE

Closes the current printer backup file so that you can print it.

SECTION 8

AUDIT TRAIL UTILITY (DMSAUD)

INTRODUCTION

The Data Management System II (DMSII) audit file print, verify, and copy utility (DMSAUD) assists you in locating errors in the audit trail and recovering from them. If the database program (DBP) fails on an audited database, the error message that appears on your terminal or the ODT shows the audit file and block affected. If you suspect a problem in the audit trail, you can check this portion of the audit for correctness using the DMSAUD utility.

The DMSAUD program can perform the following tasks:

- Produce a formatted listing of the contents of various portions of the audit trail.
- Perform various consistency checks on the audit trail records. You should perform a consistency check on each audit file before you archive it.
- Copy a secondary audit trail file to the primary audit trail, or vice versa.

INITIATING DMSAUD - SCREEN MODE

To use DMSAUD with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 4 in the YOUR CHOICE IS field on the home menu screen. Put the name of the database in the DATABASE DESIRED field; put the location (DISK, PACK, or <diskpack name>) of the control file and Data Definition File (DDF) in the ON field. Then, press the TRANSMIT key.

Figure 8-1 shows the screen that will appear on your terminal.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Audit Trail Utility (DMSAUD)

```

ACTION:      DMSAUD MENU                      10:23 A.M.      DMSUTL
              [ REFresh HOME QUIT ]
              Press SPCFY for Help

AUDIT TRAIL RANGE:

FROM:  AUDIT FILE NAME:  [      ]  TO:  AUDIT FILE NAME:  [      ]
AUDIT BLOCK NUMBER:  [      ]  AUDIT BLOCK NUMBER:  [      ]

      OR                                          OR

RELATIVE RECORD NUM:  [      ]  RELATIVE RECORD NUM:  [      ]

OPTIONS:

PRINT                      [  ]
VERIFY                     [  ]
COPY                       [  ]
SELECT SPECIFIC RECORDS    [  ]
PRIMARY AUDIT TRAIL        [  ]  OR SECONDARY AUDIT TRAIL  [  ]
```

Figure 8-1. DMSAUD Menu Screen

1. REFresh
HOMe
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. AUDIT TRAIL RANGE: The DMSAUD program processes the most recent audit file, unless you complete the FROM AUDIT FILE NAME and TO AUDIT FILE NAME fields. If you complete only the FROM AUDIT FILE NAME field, the DMSAUD program processes from that file through the end of the audit trail.
3. AUDIT BLOCK NUMBER: The DMSAUD program numbers audit blocks consecutively along the audit trail and these numbers continue across audit file boundaries. If you specify an audit block number range, do not specify a relative record number range. The block range you specify should fall within the audit files you specify. If you complete only the first AUDIT BLOCK NUMBER field, the DMSAUD program processes from that block through the last block in the ending file.

4. **RELATIVE RECORD NUM:** If you specify a relative record number range, do not specify an audit block number range. Because relative record numbers start from one (1) in each audit trail file, use the left-hand field to specify a record number in the first audit file processed, and use the right-hand field to specify a record number in the last audit file processed.

The DMSAUD program begins processing with the record that appears in the FROM AUDIT FILE NAME field and ends with the record that appears in the TO AUDIT FILE NAME field. If you fill only the FROM AUDIT FILE NAME field, the DMSAUD program continues processing through the last record in the ending file. If you want to process a single block, enter the block number in both boxes.

Select from the following options:

5. **PRINT:** When you select the PRINT option, the DMSAUD program produces a listing of the files, blocks, or records you specified in the top half of the screen. A sample copy of the report the DMSAUD program prints appears in Figure 8-5.
6. **VERIFY:** When you select the VERIFY option, the DMSAUD program checks the consistency of the data in the files, blocks, or records you specified in the top half of the screen. If the audit trail is consistent, the message "NO ERRORS IN AUDIT TRAIL" appears on the ODT.
7. **COPY:** When you select the COPY option, the DMSAUD program copies the files, blocks, or records you specify from the primary audit trail into the secondary audit trail, or vice versa. If the audit file name you specified is a primary audit trail file name (XXXXA00 through XXXXM99), or if you do not give an audit file name, but you check the "Primary Audit Trail" box, the DMSAUD program copies the data from the primary audit trail into the secondary audit trail. If you indicate a secondary audit trail file, the DMSAUD program will copy it into the primary audit trail. The DMSAUD program converts file names automatically; that is, the DMSAUD program copies file XXXXB05 into the secondary trail as XXXN05. Refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration* for more information about primary and secondary audit trails.

You can check any combination of choices 5, 6, or 7.

8. **SELECT SPECIFIC RECORDS:** When you check this field (with any nonblank character), the DMSAUD program selects only those audit trail records that meet the logical characteristics you specified and puts them into a printer backup file. The DMSUTL program displays the screen that appears in Figure 8-2, so that you can enter the logical characteristics. Do not check the PRINT, VERIFY, or COPY options, if you check this field. The records that the DMSAUD program chooses will be selected from the files, block range, or record range that you specify in the top half of the screen. If you do not specify audit file range, you may choose the audit trail from which records are to be selected by checking the PRIMARY AUDIT TRAIL OR SECONDARY AUDIT TRAIL fields.
9. **PRIMARY AUDIT TRAIL OR SECONDARY AUDIT TRAIL:** If an audit trail file name appears in one of the audit file range fields, the file name implies the primary or secondary audit trail. If you do not specify a file name, you can use these fields to indicate which audit files you want processed.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Audit Trail Utility (DMSAUD)

ACTION:	DMSAUD RECORD SELECTION MENU	10:23 A.M.	DMSUTL
	[REFresh HOME QUIT AUD-menu]		
		Press SPCFY for Help	
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []
STRUC = []	BLOCK = []	INVOC = []	TYPE []

Figure 8-2. DMSAUD Record Selection Screen

1. REFresh
HOME
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. AUD-menu: Returns you to the DMSAUD menu.

Use the DMSAUD RECORD SELECTION MENU screen to specify a total of nine types of records that you want to list. Within a line, you must complete each of the four fields before the DMSAUD program will select the record. The DMSAUD program will select a record if it meets the criteria given in any one line. That is, there is an AND relationship between fields in a line and an OR relationship between lines. The DMSUTL program does not require leading zeros in these fields.

5. STRUC: This is the structure number from which the DMSAUD program selects records. These records will pertain to database activity on this database structure.
6. BLOCK: The DMSAUD program will select records that pertain to activity on this block of the structure that you specified.
7. INVOC: The DMSAUD program will select records that pertain to the invocation number of the structure you specified.
8. TYPE: The DMSAUD program will select records that pertain to this type of database transaction. You may enter either the <type code> or the <mnemonic> (see Table 8-1).

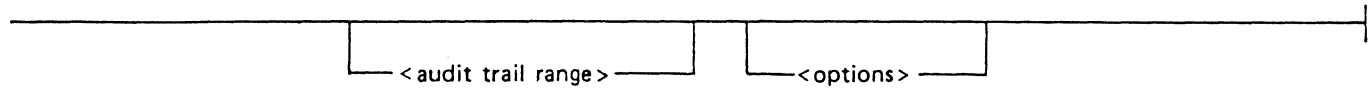
Table 8-1. Audit Record Types

<type code>	<mnemonic>
1	SYNC-POINT
2	CONTROL-POINT
3	DB-OPEN
4	DB-CLOSE
5	DBP-OPEN
6	DBP-CLOSE
7	AUDIT-FILE-CHANGE
8	RECOVERY
9	AUDIT-DISCONTINUITY
10	DS-STORE-NEW
11	DS-DELETE
12	DS-UPDATE
14	DS-BCI-BEFORE
15	DS-BCI-AFTER
16	RND-DS-STORE-NEW
17	RND-DS-DELETE
20	SET-ADD-ENTRY
21	SET-DELETE-ENTRY
22	SET-UPDATE-ENTRY
24	SET-TABLE-SPLIT
25	IS-ROOT-SPLIT
26	SET-BCI-BEFORE
27	SET-BCI-AFTER
30	BLOCK-ONE-UPDATE
31	GENERAL-BEFORE-IMAGE
32	GENERAL-AFTER-IMAGE
33	DATA-WRITE-ERROR
34	DATA-READ-ERROR
35	INIT-RND-BLOCK
40	PURGE-TABLE
41	PURGE-DONE
42	START-OF-DUMP
43	END-OF-DUMP
80	BEGIN-TRANSACTION
81	END-TRANSACTION

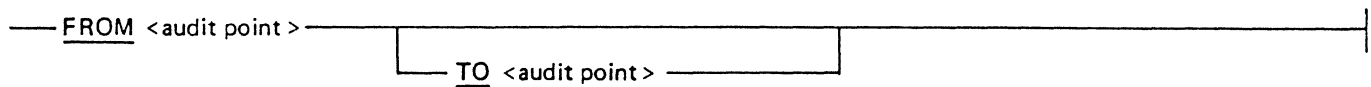
FREEFORM SYNTAX

The freeform syntax for DMSAUD appears in Figure 8-3. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSAUD screen. For further discussion of the options, refer to the "Initiating DMSAUD - Stand-Alone Mode" portion of this section. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

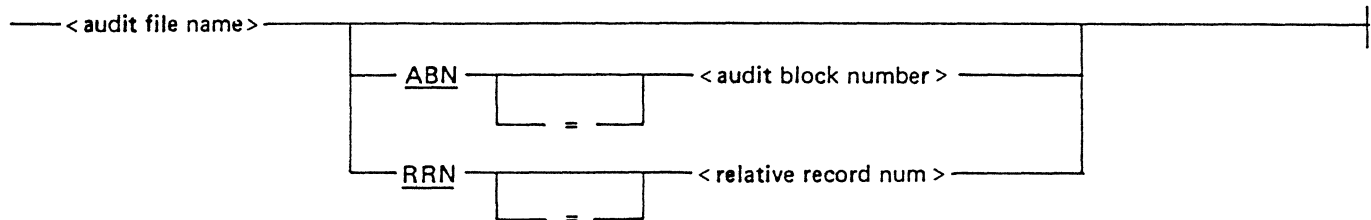
<dmsaud free form syntax parameters>



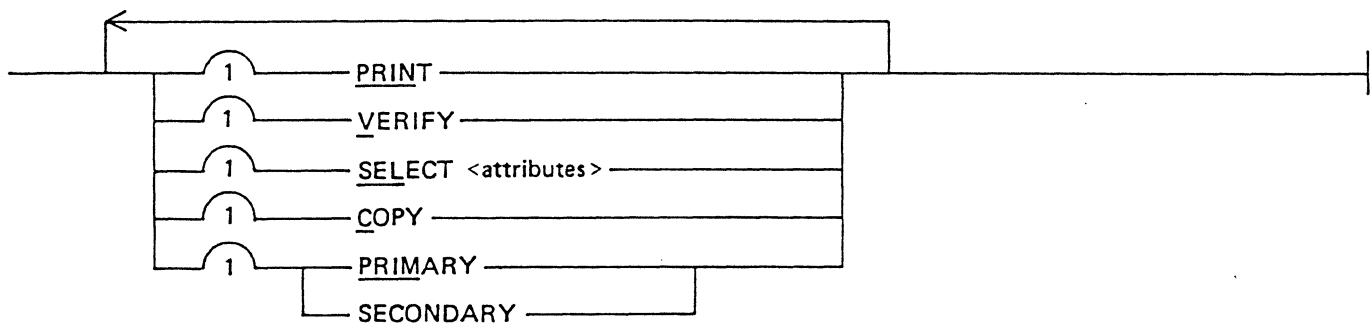
<audit trail range>



<audit point>



<options>



<attributes>

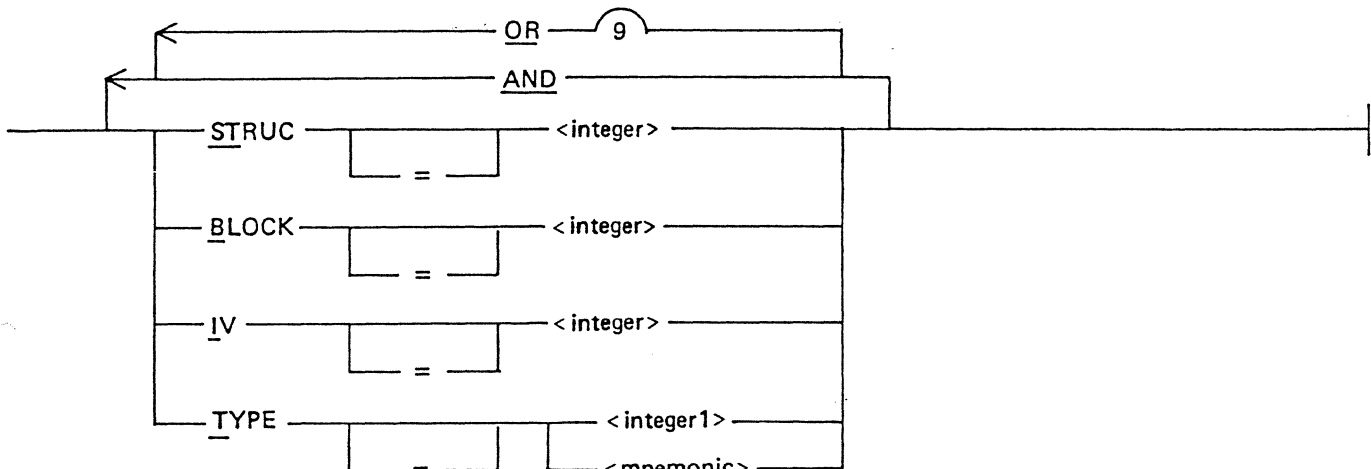


Figure 8-3. DMSUTL Freeform Syntax for DMSAUD

INITIATING DMSAUD - STAND-ALONE MODE

You cannot initiate DMSAUD with the DMSUTL batch mode. However, because DMSAUD is an independent program, you can initiate it from the ODT, from cards or card images, or from a WFL job.

From the ODT

You can execute DMSAUD by entering the following syntax at the ODT:

```
EX DMSAUD
DATA AUDIN
<parameters>
END
      OR
EX DMSAUD/SPO
<mix no.> AX "<parameters>"
END
```

Figure 8-4 shows the syntax for <parameters>.

From Cards or Card Images

You can execute DMSAUD by entering the following syntax in a card deck or pseudodeck:

```
? EX DMSAUD
? DATA AUDIN
  <parameters>
? END
```

Figure 8-4 shows the syntax for <parameters>.

From a WFL Job

You can initiate DMSAUD by entering the following WFL statements:

```
RUN DMSAUD
DATA AUDIN
<parameters>
!
```

Figure 8-4 shows the syntax for <parameters>.

PARAMETERS

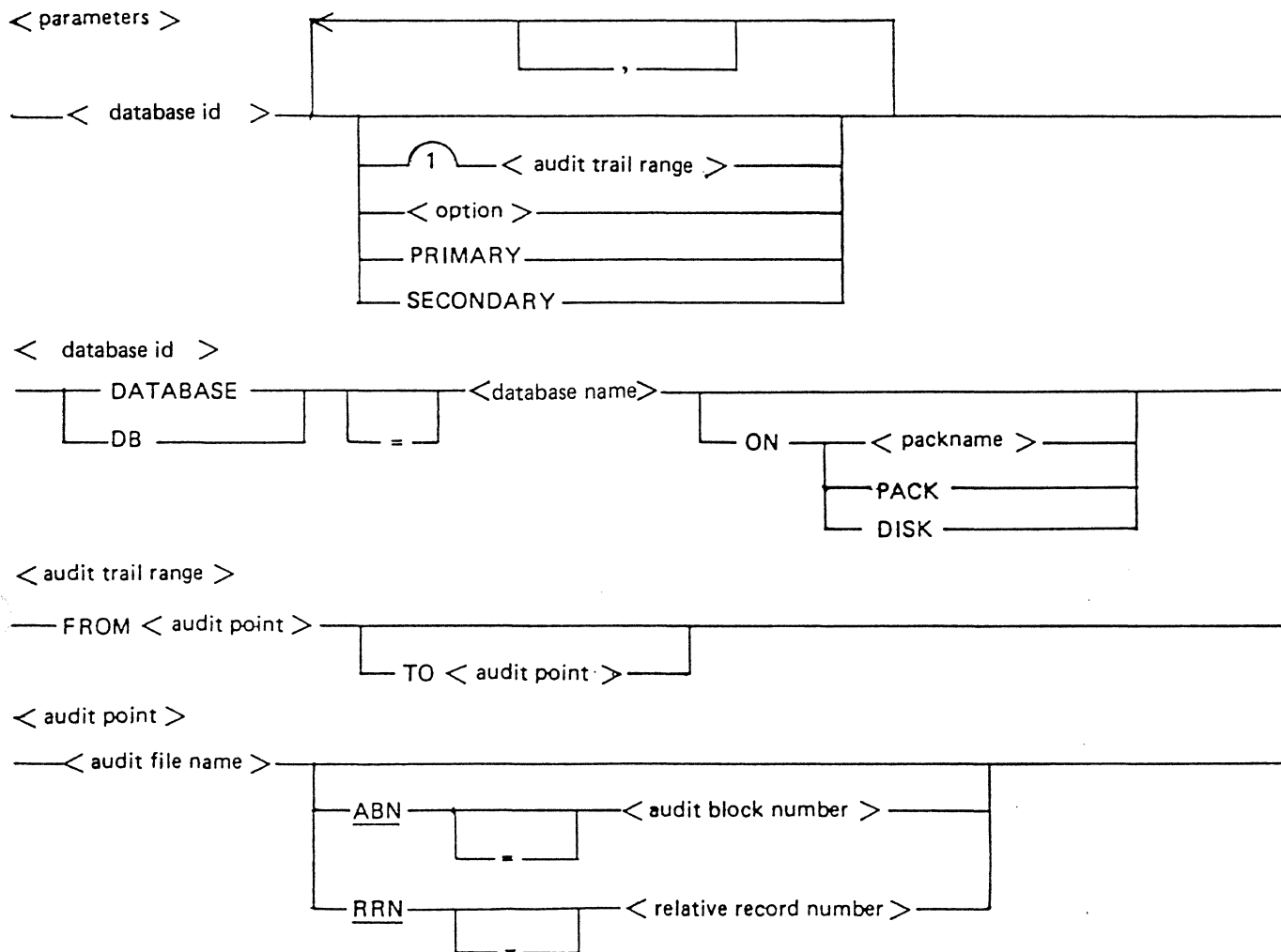


Figure 8-4. DMSAUD Parameters for Batch Use (Sheet 1 of 2)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Audit Trail Utility (DMSAUD)

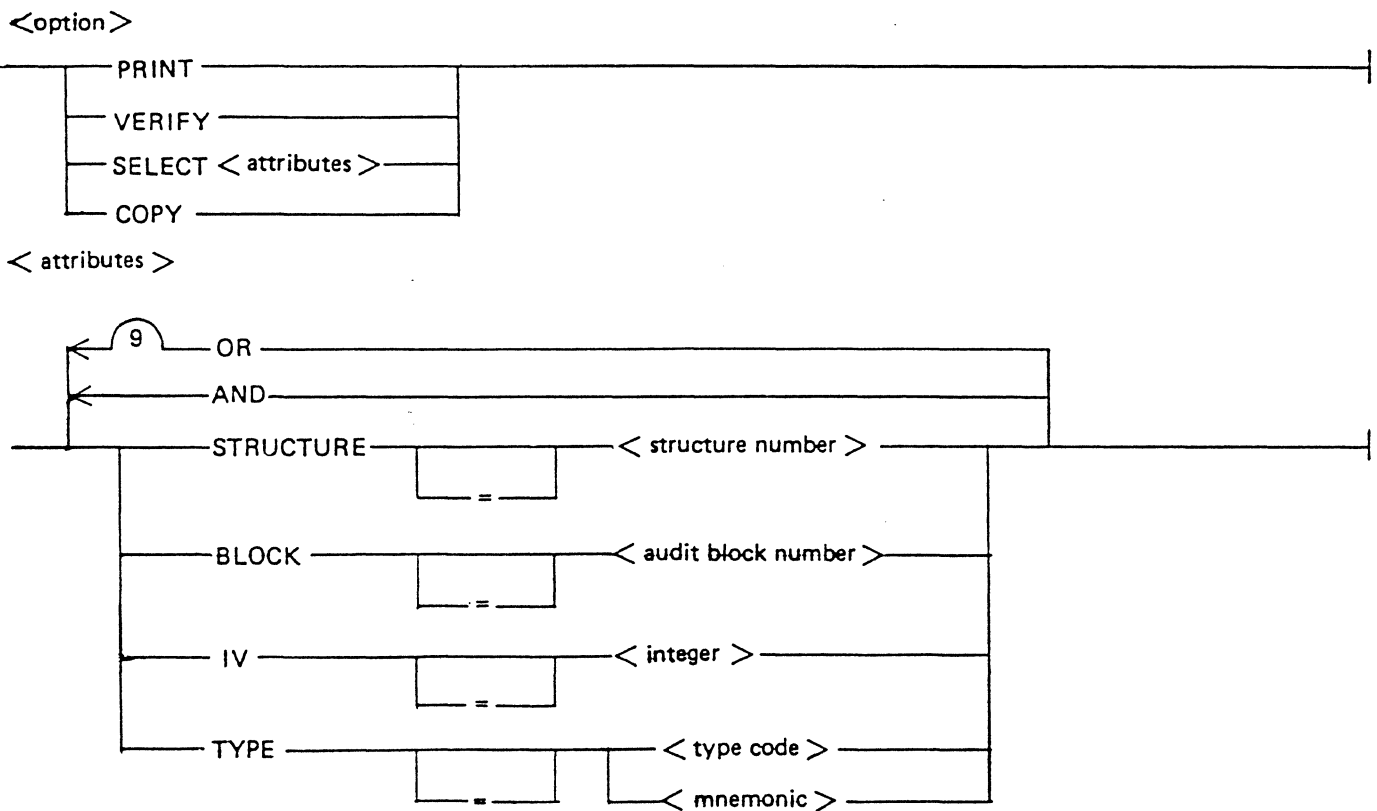


Figure 8-4. DMSAUD Parameters for Batch Use (Sheet 2 of 2)

<audit trail range>

The <audit trail range> specifies the beginning and ending point in the audit trail for the option you select. If you omit the ending point, the DMSAUD program assumes that you want to process to the end of the audit trail. If you omit the audit trail range, the DMSAUD program uses only the last audit file.

PRIMARY

SECONDARY

Use PRIMARY or SECONDARY to select either the primary or the secondary (duplicated) audit trail that you want the DMSAUD program to report on. This is only valid if there is a duplicated audit trail (that is, if you specified DUPLICATE in the audit trail description in DASDL).

<identifier>

Identifier is the name (3 to 6 characters) of the database.

ON PACK

ON DISK

ON <packname>

Use this option to specify the location of the DDF.

PRINT

When you select the PRINT option, the DMSAUD program prints the contents of the audit trail. Also, the DMSAUD program lists the content of each audit record with the appropriate headings and format, according to the type, lists Block Control Information, and lists Block one information.

VERIFY

When you select the VERIFY option, the DMSAUD program performs various consistency checks on the audit trail and generates a printout identifying the records or blocks found in error.

SELECT

The SELECT option enables you to request reports for only certain audit trail records that satisfy given attributes. Do not use the SELECT option with PRINT, COPY, or VERIFY.

COPY

When you select the COPY option, the DMSAUD program copies an audit trail file or files from the secondary to the primary audit trail, or from the primary to the secondary audit trail. Use this if a user program performing an update has turned on DUPLICATED audit and the programmer wishes to perform rollback recovery to a point prior to the update. Also, use the COPY option if the audit trail file is lost. The DMSAUD program performs an unconditional VERIFY before the copy and performs any PRINT option you have requested before it performs the COPY function.

If you do not specify an option, the DMSAUD program uses the default value of VERIFY. If you specify both the PRINT and VERIFY options, the DMSAUD program lists any error messages following the record, block, or file information found in error.

When the DMSAUD program completes the VERIFY or PRINT operation, it may try to OPEN an audit trail file which does not exist. If this happens, a message such as "NO FILE DBTA07" will appear on the ODT. At this point, you must use the OF command to specify that the file is optional and that DMSAUD should terminate. For more information about the OF command, see the *B 2000/B 3000/B 4000/V Series MCP/IX System Software Operations Guide Volume 1: System Commands* or the *V Series MCP/VS System Software Operations Guide Volume 2: System Commands*.

<attributes>

Use attributes to select the part of the audit trail that you want the DMSAUD program to report on.

AND

OR

Use AND and OR in the logical sense. The DMSAUD program allows a maximum of nine OR cases.

STRUC

When you specify STRUC, the DMSAUD program reports on audit records pertaining to this structure. The DMSCTL program produces a report that shows the structure number.

BLOCK

Use BLOCK to select an Audit Block Number (ABN) in the audit trail.

TYPE

Use TYPE to select the types of audit records that you want the DMSAUD program to report on. The audit record types appear in Table 8-1.

Example

```
SELECT TYPE = DS-UPDATE AND STRUC = 3 AND  
      BLOCK = 200 OR STRUC = 4 AND BLOCK = 32
```

With the syntax in the example, the DMSAUD program will find and print all of the data set update records for structure 3 that apply to block 200. In addition, it will find and print all of the audit trail records for structure 4 that apply to block 32, regardless of type.

DMSAUD REPORTS

Samples of DMSAUD reports appear in Figure 8-5. The following terms appear in the reports:

1. DMSII AUDIT TRAIL FOR DATABASE: <database name>
2. INPUT PARAMETERS TO DMSAUD: The parameters that were received to produce this report.
3. AUDIT FILE: <audit file name>
4. BLOCK ONE INFORMATION: Information in the control block of this audit file
 - a. DBNAME: name of the database
 - b. FILE: name of the physical audit file
 - c. EOF: current End Of File pointer for the audit file
 - d. BEG-ABN: beginning Audit Block Number in the audit file
 - e. END-ABN: ending Audit Block Number in the audit file
 - f. P: prior I/O error (Boolean value)
 - g. R/D: prior I/O error result descriptor
 - h. FL: format level of file
 - i. C: checksum (Boolean value)
5. BLOCK TRAILER INFORMATION: Information in the BCI for this audit block
 - a. FILE BLOCK: block number of this block in this file
 - b. TRAIL BLOCK: Audit Block Number (ABN) of this block in the audit trail
 - c. ABN: audit block number
 - d. F: first record in this block is split (Boolean value)
 - e. L: last record in this block is split (Boolean value)
 - f. F-BCW: block offset of first Begin Control Word
 - g. L-BCW: block offset of last Begin Control Word

- h. L-USED: block offset of next available space in this block
 - i. DATE-TIME-STAMP: date and time this audit block was written
 - j. CHECKSUM: value of checksum for this block
 - k. TIME: break out of time from DATE-TIME-STAMP in hh:mm:ss:mmm
 - l. DATE: break out of date from DATE-TIME-STAMP in mm/dd/yy
 - m. WEEKDAY: break out of day of week from DATE-TIME-STAMP
- 6. Audit records in block edited by format
 - 7. Items 5-6 repeated for each block in the audit file
 - 8. Items 3-7 repeated for each file in the <audit trail range>
 - 9. A line of "***I/O ERROR**I/O ERROR**" followed by ABN and expected Audit Serial Number (ASN) for any blocks containing I/O errors

More information concerning the fields for the various record types appears in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

A sample output appears in Figure 8-5.

DMSII AUDIT TRAIL FOR DATABASE: ADMIN ON THINGS

INPUT PARAMETERS TO DMSAUD

DATABASE = ADMIN ON THINGS FROM ADMA04 TO ADMA04 PRINT END ← Selected by user

AUDIT FILE: ADMA04 ON PACK

BLOCK ONE INFORMATION

DBNAME FILE EOF BEGIN-ABN END-ABN P R/D FL C
ADMIN ADMA04 00000005 000000000156 000000000159 0 000 03 1

BLOCK 1 TRAILER INFORMATION

ABN F L F-BCW L-BCW L-USED DATE-TIME-STAMP CHECKSUM TIME DATE WEEKDAY
000000000000 0 0 000000 000000 000000 872730058132621 33158130C093 16:08:52:621 09/30/87 (WEDNESDAY)

==== START OF BLOCK 00000002 =====

BLOCK TRAILER INFORMATION

FILE BLOCK: 00000002 TRAIL BLOCK: 000000000156

ABN F L F-BCW L-BCW L-USED DATE-TIME-STAMP CHECKSUM TIME DATE WEEKDAY
000000000156 0 0 000000 000000 000063 872730058134077 1201104F0510 16:08:54:077 09/30/87 (WEDNESDAY)
BCW: TYPE=RECOVERY OFFSET=000000

Begin control word

TP SIZE RT SP-LINK(FILE:BLOCK:OFFSET) DATE-TIME-STAMP
08 000063 02 ADMA03:000000000155:000000 872730058134077
[02] := HALTLOAD

TP = record type

ECW: TYPE=RECOVERY SIZE=000063

End control word

==== START OF BLOCK 00000003 =====

BLOCK TRAILER INFORMATION

FILE BLOCK: 00000003 TRAIL BLOCK: 000000000157

ABN F L F-BCW L-BCW L-USED DATE-TIME-STAMP CHECKSUM TIME DATE WEEKDAY
000000000157 0 0 000000 000747 000767 872730058188262 928A8F0FF37B 16:09:48:262 09/30/87 (WEDNESDAY)

Figure 8-5. Sample Listing from DMSAUD (Sheet 1 of 4)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Audit Trail Utility (DMSAUD)

BCW: TYPE=DBP OPEN

OFFSET=000000

TP SIZE DATE-TIME-STAMP
05 000031 872730058163969

TIME DATE WEEKDAY
16:09:23:969 09/30/87 (WEDNESDAY)

ECW: TYPE=DBP OPEN SIZE=000031

BCW: TYPE=DB OPEN

OFFSET=000031

TP SIZE PROG IV TC DATE-TIME-STAMP
03 000047 ADMLOD 03 01 872730058163970

TIME DATE WEEKDAY
16:09:23:970 09/30/87 (WEDNESDAY)

ECW: TYPE=DB OPEN SIZE=000047

BCW: TYPE=BEGIN TRANSACTION

OFFSET=000078

TP SIZE IV CT
80 000020 03 01

ECW: TYPE=BEGIN TRANSACTION SIZE=000020

BCW: TYPE=SET ADD ENTRY

OFFSET=000098

TP SIZE STRU IV PACN(ABN:OFFSET) BLOCK SLOT R H MOVE
20 000072 0016 03 000000000153:003304 00000002 0248 0 0 0001

FIRST IMAGE
0000040000 0024

ECW: TYPE=SET ADD ENTRY SIZE=000072

BCW: TYPE=SET ADD ENTRY

OFFSET=000170

TP SIZE STRU IV PACN(ABN:OFFSET) BLOCK SLOT R H MOVE
20 000072 0017 03 000000000153:002754 00000002 0249 0 0 0001

FIRST IMAGE
0000040000 0024

ECW: TYPE=SET ADD ENTRY SIZE=000072

Figure 8-5. Sample Listing from DMSAUD (Sheet 2 of 4)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Audit Trail Utility (DMSAUD)

BCW: TYPE=DS STORE NEW

OFFSET = 000242

TP SIZE STRU IV PACN(ABN:OFFSET) BLOCK SLOT
10 000124 0013 03 000000000153:003376 00000002 0002

FIRST IMAGE

0000000300 0004C4C1E3 C140C3D3C5 D9D24040E3 0249236607 4411160000 000000007F 23

ECW: TYPE=DS STORE NEW

SIZE=000124

BCW: TYPE=DS UPDATE

OFFSET = 000366

TP SIZE STRU IV PACN(ABN:OFFSET) BLOCK SLOT
12 000268 0003 03 000000000153:003500 00000002 0000

FIRST IMAGE

E3D9C1C3C5 4000000000 0000000301 0100000000 0002E3D9C1 C3C5400000 0000000040 4040404040 4040404040 4040404040
40404040

SECOND IMAGE

E3D9C1C3C5 4000000000 0000000401 0100000000 0003E3D9C1 C3C5400000 0000000040 4040404040 4040404040 4040404040
40404040

ECW: TYPE=DS UPDATE

SIZE=000268

BCW: TYPE=END TRANSACTION

OFFSET = 000634

TP SIZE IV CT
81 000020 03 00

ECW: TYPE=END TRANSACTION

SIZE=000020

BCW: TYPE=SYNC POINT

OFFSET = 000654

TP SIZE CP-LINK(FILE:BLOCK:OFFSET)
01 000046 ADMA04:000000000157:000000

ECW: TYPE=SYNC POINT

SIZE=000046

Figure 8-5. Sample Listing from DMSAUD (Sheet 3 of 4)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Audit Trail Utility (DMSAUD)

BCW: TYPE=DB OPEN

OFFSET = 000700

TP SIZE PROG IV TC DATE-TIME-STAMP
03 000047 ADMWRK 08 02 872730058188262

TIME	DATE	WEEKDAY
16:09:48:262	09/30/87	(WEDNESDAY)

ECW: TYPE=DB OPEN SIZE=000047

BCW: TYPE=BEGIN TRANSACTION

OFFSET = 000747

TP SIZE IV CT
80 000020 08 01

ECW: TYPE=BEGIN TRANSACTION SIZE=000020

==== START OF BLOCK 00000004 =====

BLOCK TRAILER INFORMATION

FILE BLOCK: 00000004 TRAIL BLOCK: 000000000158

ABN	F	L	F-BCW	L-BCW	L-USED	DATE-TIME-STAMP	CHECKSUM	TIME	DATE	WEEKDAY
000000000158	0	0	000000	000000	000063	872730058189647	7D3CA32FF0DB	16:09:49:647	09/30/87	(WEDNESDAY)

BCW: TYPE=RECOVERY

OFFSET = 000000

SIZE RT SP-LINK (FILE:BLOCK:OFFSET) DATE-TIME-STAMP
08 000063 01 ADMA04:000000000157:000654 872730058189647
[01] := ABORT

TIME	DATE	WEEKDAY
16:09:49:647	09/30/87	(WEDNESDAY)

ECW: TYPE=RECOVERY SIZE=000063

==== START OF BLOCK 00000005 =====

BLOCK TRAILER INFORMATION

FILE BLOCK: 00000005 TRAIL BLOCK: 000000000159

ABN	F	L	F-BCW	L-BCW	L-USED	DATE-TIME-STAMP	CHECKSUM	TIME	DATE	WEEKDAY
000000000159	0	0	000000	000690	000721	872730058229084	722C8E2B079A	16:10:29:084	09/30/87	(WEDNESDAY)

Figure 8-5. Sample Listing from DMSAUD (Sheet 4 of 4)

DESCRIPTION OF CONSISTENCY ANALYSIS

When you specify the VERIFY option, the DMSAUD program analyzes the audit trail for the following conditions.

For Block One:

- The audit file name and database name fields must contain the expected values.
- The logical End Of File (EOF) pointer must point to the last block in the file.
- The beginning and ending ABN values must be correct.

For all other blocks in the audit file:

- The Audit Block Numbers must be in ascending numerical sequence.
- The split record flags must be consistent.
- The date-timestamps must be equal or ascending (changing only at allowed points).
- The computed checksum value must match.
- The first and last Block Control Word (BCW) offsets must be correct.

For the audit records:

1. The record size must be correct.
2. The begin and end control words must match.
3. The syncpoint and controlpoint linkages must be correct.
4. The record date-timestamps must be in ascending sequence.

SECTION 9

SOURCE GENERATION UTILITY (DMSGEN)

INTRODUCTION

The DMSGEN utility generates V Series COBOL ANSI-74 source code for three kinds of programs.

- Programs generated by the UNLOAD function of DMSGEN input Data Management System II (DMSII) database files and output regular, "flat" disk or diskpack files containing the same information. You can use the flat files as input for programs you write that produce reports, for example. These programs can read the flat files more quickly than they could access the database files through DMSII, and processing can be done offline, so that it does not interfere with database functioning.
- Programs generated by the LOAD function of DMSGEN input ordinary disk or diskpack files and load their information into a DMSII database. The files may be those prepared by the UNLOAD program or may be any file whose records "map onto" the DASDL description of the database records. This is a quick way to convert information to DMSII format.
- The SKELETON function of DMSGEN generates the basic skeleton of a V Series COBOL ANSI-74 database user program. The code generated is specific to a particular database you named. It includes recovery and restart logic with maintenance of a restart data set, and handling of error reporting and DMERROR USE procedures. You can insert the actual FIND and STORE logic and compile the program. In this way, the DMSGEN program automatically handles some of the more intricate and tedious coding tasks of DMSII user programming.

The following paragraphs provide more information about these functions. The source code that the DMSGEN program generates contains comments to explain the program logic and to highlight important spots in the program. This manual does not include examples of code listings because they vary with the form of the database you specify.

INITIATING DMSGEN - SCREEN MODE

To use DMSGEN with the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 7 in the YOUR CHOICE IS field on the home menu screen. Then, press the TRANSMIT key.

Figure 9-1 shows the screen that will appear on your terminal.

```
ACTION:  DMSGEN MENU                10:23 A.M.      DMSUTL
          [                      ]
          LOAD UNLoad SKEleton
          REFresh HOMe QUIT          Press SPCFY for Help

-----

OPTIONS AVAILABLE:

        DEBUG  [ ]
        RESTART [ ]

NAME OF RESTART DATA SET:  [          ]

-----
```

Figure 9-1. DMSGEN Menu Screen

1. REFresh
HOMe
QUIT
For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.
2. LOAd: Generates a recoverable COBOL program that loads information from conventional flat files into the database.
3. UNLoad: Generates a recoverable COBOL program that unloads information from database files into conventional flat files.
4. SKEleton: Generates a restartable COBOL skeleton program.
5. DEBUG: When you check this field, the database program (DBP) sets the DEBUG option in the COBOL user program. The default value is NON-DEBUG.
6. RESTART: If you check this field, enter the restart data set name. If you do not enter a name, the DBP uses the name: "RESTART-FOR-LOAD."

FREEFORM SYNTAX

The freeform syntax for DMSGEN appears in Figure 9-2. You can enter this syntax on the bottom of the home menu screen. The options correspond to the fields and choices on the DMSGEN screen. For further discussion of the options, refer to the "Initiating DMSGEN - Batch Mode" portion of this section. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

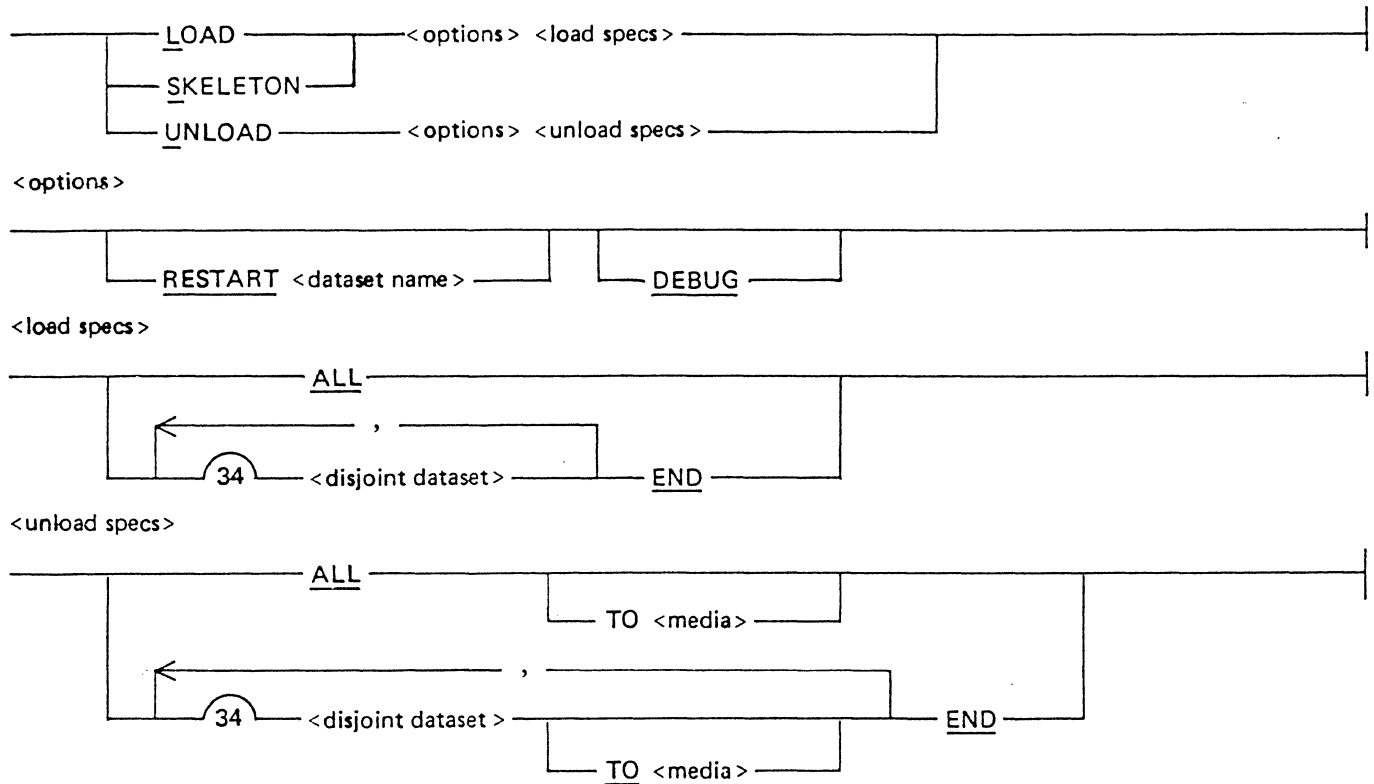


Figure 9-2. DMSUTL Freeform Syntax for DMSGEN

INITIATING DMSGEN - BATCH MODE

For general directions on the use of batch mode, refer to the "Using DMSUTL in Batch Mode" portion of Section 1. Figure 1-3 shows the general syntax; Figures 9-3 and 9-4 show the DMSGEN specification syntax.

< dmsgen specs >

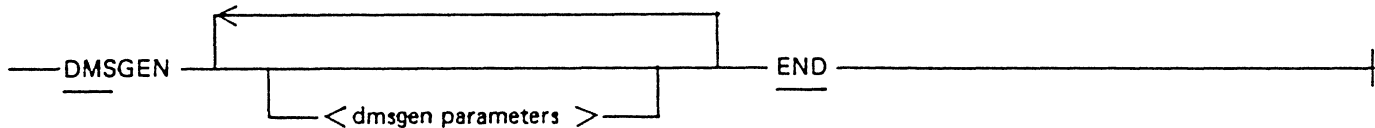
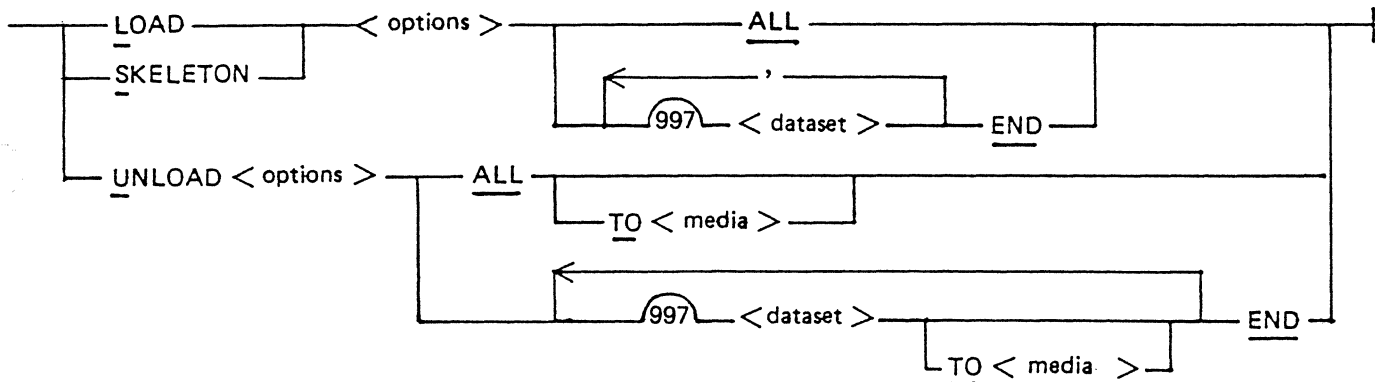


Figure 9-3. DMSGEN Specification for Batch Use

< dmsgen parameters >



< options >

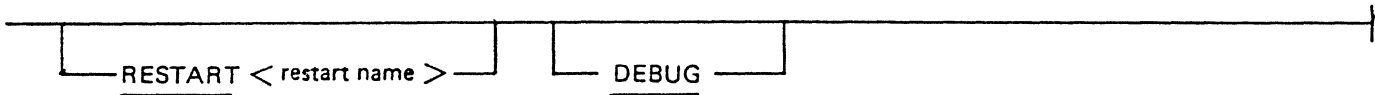


Figure 9-4. DMSGEN Parameters for Batch Use

LOAD

With this parameter, the DMSGEN program generates a COBOL program that loads data into the database.

UNLOAD

With this parameter, the DMSGEN program generates a COBOL program that unloads data from the database.

SKELETON

With this parameter, the DMSGEN program generates the skeleton of a COBOL database user program.

DEBUG

With this option, the DBP adds debugging code to the skeleton COBOL user program. The default is NON-DEBUG.

RESTART

With this option, you can specify the data set name to be used for the restart data set. The default is "RESTART-FOR-LOAD."

<restart name>

The restart data set name can be a maximum of 17 characters.

ALL

With this parameter, all structures are included in the generated program.

<data set>

This must be the name of a disjoint data set and it is a maximum of 17 characters. Input may extend over several cards or AX commands (ACCEPT) until you enter "END". The generated program will include only these data sets.

UNLOAD FUNCTION

The UNLOAD function produces V Series COBOL ANSI-74 source for a program that will input DMSII database files and produce regular, "flat" disk or diskpack files containing the same information. You can use these files for any function needed by the installation. It is quicker to read a flat file than it is to access information from a database. Also, you can process these files offline while a user program performs an update of the database. It is not recommended that you use this function for archiving database information; you should use the DUMP and LOAD options available with the DMSDLR utility to archive the database.

The UNLOAD program, as generated, unloads only disjoint data sets. The programmer can add logic to also unload embedded data sets, disjoint manual subsets, embedded manual subsets, and automatic subsets, if needed.

Generating an UNLOAD Program

When you type UNL in the ACTION field of the DMSGEN menu, the DMSUTL utility displays a screen that contains fields so that you can enter the names of the structures you want unloaded. You can request that the DBP unload to DISK or PACK. If there is not enough room on the screen to enter the names of all the structures, enter as many as will fit on this screen and check the CHECK IF MORE DATA SETS REQUIRED field. This process can be repeated as many times as is needed to enter up to 997 structures.

You can enter only disjoint data sets on these screens. If you need to unload other structures, the programmer must customize the COBOL source code produced by DMSGEN.

Also, you can generate an UNLOAD program with freeform syntax or with the batch mode syntax.

Format of Flat Files

DISJOINT DATA SETS

The UNLOAD program produces one flat file from each disjoint data set you specified on the "DMSGEN DATA SET MENU" screen. Each record in the flat file will contain the data from one record in the corresponding data set. The following are the attributes of the flat file that the UNLOAD program creates:

External file name: UNL<structure number>. (For example, UNL004)

Internal file name: <data set name>-FILE. (For example, BANKS-FILE)

Record length: This is the record length you declared in DASDL for structure plus 8 digits for a Relative Record Number (RRN).

Blocking: Blocked the same as the data set.

LOAD FUNCTION

The LOAD function produces V Series COBOL ANSI-74 source for a program that will input regular disk or diskpack files and load the information into a DMSII database. The LOAD program loads the information into a database more quickly than a regular user program could do it. All that is necessary is that the record format of the files be congruent with the corresponding database record format that you declared in DASDL.

The LOAD program, as generated, loads only disjoint data sets. The programmer can add logic to also load embedded data sets, disjoint manual subsets, embedded manual subsets, and automatic subsets, if needed.

Generating a LOAD Program

Type LOA in the ACTION field on the DMSGEN menu. The DMSUTL program displays a screen with fields so you can enter the names of data sets that you want loaded. If you need to load more structures than will fit on the screen, enter as many as will fit on this screen and check the CHECK IF MORE DATA SETS REQUIRED field. This process can be repeated as many times as is needed to enter up to 997 structures.

You can enter only disjoint data sets on these screens. If you need to load other structures, the programmer must customize the COBOL source code that DMSGEN produces.

Also, you can generate a LOAD program with freeform syntax or with batch mode syntax.

The program produced by the LOAD function of DMSGEN expects the flat files to be on disk or diskpack. It expects them to be called UNL<structure number> - for example, UNL004 for structure 4. If you cannot meet these expectations, the programmer must customize the COBOL source code that DMSGEN produces.

SKELETON FUNCTION

The SKELETON function produces V Series COBOL ANSI-74 source for part of a DMSII user program. The parts produced handle restart and error recovery functions, maintain a restart data set file, and produce error messages. This function also produces file descriptions. You then add the logic for the actual database reads and writes.

Generating a SKELETON Program

Type SKE in the ACTION field on the DMSGEN menu screen. The DMSUTL program displays a screen with fields in which you enter names of data sets. Each of these structures will become a level 01 data item in the file declaration of the skeleton program. If you need to add more items than will fit on the screen, enter as many as will fit on this screen and check the CHECK IF MORE DATA SETS REQUIRED field. This process can be repeated as many times as is needed to enter up to 997 structures.

Also, you can generate a SKELETON program with freeform syntax or with batch mode syntax.

You can enter only data set names on these screens. If you need other structures in the file description, the programmer must customize the COBOL source code that DMSGEN produces.

SECTION 10

DATABASE MAINTENANCE UTILITY (DMSDLR)

INTRODUCTION

DMSDLR is a database utility program that performs several database maintenance functions (Dump, Load, Recovery, etc.). You can initiate these functions by using the IR command, by executing DMSDLR directly, or by using the DMSUTL utility. This section describes the DMSDLR functions and how to initiate them using the DMSUTL utility:

- **Recovery:** The recovery functions the DMSUTL utility enables you to initiate (HALT/LOAD, REBUILD, ROLLBACK, and RECONSTRUCT) can return a database to a consistent, usable condition after a failure of the database program (DBP), the operating system, or the hardware.
- **Reorganization:** The REORGANIZE function enables you to make the following types of changes to database structure files: record format changes, file attribute changes, garbage collection, set balancing/generate-new, and subset balancing/generate-new. Refer to the Reorganize Function portion of this section for more information about generate-new.
- **Production:** The PRODUCTION function enables you to initiate a production database program. The Production DBP must already be bound and on disk. A production DBP is a database user program bound in with a DBP to optimize its performance.
- **Purge:** The PURGE function enables you to remove all the user data from database structure files. You cannot do a purge when user programs are accessing the database and you cannot purge a restart data set.
- **Load:** The LOAD function enables you to load the entire database from a full database backup tape. The database is left in a ready-to-run state, and recovery is not allowed.
- **Dump:** The DUMP function enables you to create database backup tapes. These tapes can then be used in later LOAD or recovery functions if these become necessary.
- **Restart:** The RESTART function enables you to restart when a nonerror failure occurs during one of the following functions: REBUILD, RECONSTRUCT, DATABASE LOAD, or REORGANIZATION.

The following paragraphs provide a more detailed discussion of each of these DMSDLR functions.

INITIATING DMSDLR FUNCTIONS - SCREEN MODE

To use the DMSDLR functions of the DMSUTL utility, initiate DMSUTL with CANDE, GEMCOS, or MC7 as described in Section 1. Put a 9 in the YOUR CHOICE IS field on the home menu screen. Then, press the TRANSMIT key.

Figure 10-1 shows the screen that will appear on your terminal.

ACTION:	DMSDLR MENU	10:23 A.M.	DMSUTL
	[		
	REFresh	HOME	QUIT
		Press SPCFY	for Help
FUNCTIONS AVAILABLE:			
0.	HALTLOAD	5.	PRODUCTION
1.	REBUILD	6.	PURGE
2.	ROLLBACK	7.	LOAD
3.	RECONSTRUCT	8.	DUMP
4.	REORGANIZE	9.	RESTART
YOUR CHOICE IS: []			

Figure 10-1. DMSDLR Menu Screen

1. REFresh
HOME
QUIT

For an explanation of these selections, see the "Using the DMSUTL Data Entry Screens" portion of Section 1.

2. FUNCTIONS AVAILABLE: The following paragraphs provide a discussion of each function.
3. YOUR CHOICE IS: Enter 0-9 and press the TRANSMIT key. The DMSUTL utility displays the appropriate menu screen.

HALT/LOAD FUNCTION

You use halt/load recovery after a system failure and subsequent halt/load of the MCP, or after a DBP failure. A halt/load recovery can be performed only on an audited database. To initiate halt/load recovery enter a 0 in the YOUR CHOICE IS field of the DMSDLR menu screen and press the TRANSMIT key.

For more information about halt/load recovery, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

REBUILD FUNCTION

Rebuild recovery loads a backup copy of the database and recovers it forward in time to the point you specify by applying transaction records from the audit trail. It determines the starting point in the audit trail by referring to the reloaded control file. All of the database files are recovered to the point you specify. Rebuild recovery can be performed only on an audited database.

You must reload all the audit files created since the database backup tape was made before you initiate a rebuild recovery. You can use the SYSTEM/COPY utility to do this.

The rebuild recovery process applies audit trail records until it reaches the stopping point you specify in the audit trail. You can specify the stopping point in one of five ways:

- The beginning of an audit trail file. You enter the audit file name as XXXLNN where XXX represents the first three characters of the database name; L represents a letter from A-M for the primary audit trail and N-Z for the secondary audit trail; and NN represents a two-digit number.
- The first Beginning Of Job (BOJ) or End Of Job (EOJ) audit record encountered from a specific user program. The rebuild process will not apply the BOJ or EOJ record. You can use this stopping option to restore the database to a point in time just prior to the execution of a particular database user program that has compromised the database. Enter the user program name (up to 6 characters).
- An audit record created at or before a specified date and time. The rebuild process will not apply the record that meets the criteria for the specified date and time. If you know the time of the corrupting action, you can use this stopping option to restore the database to a point in time just before that action. Enter the date and time.

You can enter month as, for example, FEBRUARY or FEB. Day can be one or two digits. Year is optional, and can be two or four digits. Hour is one or two digits on a 24-hour clock. Minutes can be one or two digits. Seconds are optional, and can be one or two digits. Milliseconds are optional, and can be up to three digits.

- A specific record in the audit trail. You specify the location of the record in terms of Audit Block Number (ABN) and digit offset within that audit block. Audit Block Numbers are absolute, starting at 1 at the beginning of the audit trail (not starting at 1 for each audit file). Audit Block Numbers and offsets can be up to 12 digits long.
- The end of the audit trail. This stopping option restores the database to its most recent state by applying all the audit records in the audit trail. You select this option by not specifying a stopping point.

You can load the backup tape automatically by specifying its name in the appropriate field and mounting the tape. The rebuild process will then load the tape and begin the recovery.

If a nonerror failure occurs during a rebuild recovery, you can restart the failed function with the RESTART function.

For more information about rebuild recovery, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

ROLLBACK FUNCTION

Rollback recovery takes a current database and recovers it backward in time by reading back through the audit trail and applying before images to back out transaction records from the database. All of the database files are rolled back to the point you specify. Rollback recovery can only be performed on audited databases.

You must reload all the audit files created since your chosen stopping point in the audit trail before you initiate rollback recovery. You can use the SYSTEM/COPY utility to do this.

The rollback recovery process applies audit trail records moving backwards until it reaches the stopping point you specify in the audit trail. You can specify the stopping point in one of four ways:

- The beginning of an audit trail file. You enter the audit file name as XXXLNN where XXX represents the first three characters of the database name; L represents a letter from A-M for the primary audit trail and N-Z for the secondary audit trail; and NN represents a two-digit number.
- The first Beginning Of Job (BOJ) or End Of Job (EOJ) audit record encountered from a specific user program. The rollback process will not apply the BOJ or EOJ record. You can use this stopping option to restore the database to a point in time just prior to the execution of a particular database user program that has compromised the database. Enter the user program name (up to 6 characters).
- An audit record created at or before a specified date and time. The rollback process will not apply the record that meets the criteria for the specified date and time. If you know the time of the corrupting action, you can use this stopping option to restore the database to a point in time just before that action. Enter the date and time.

You can enter month as, for example, FEBRUARY or FEB. Day can be one or two digits. Year is optional, and can be two or four digits. Hour is one or two digits on a 24-hour clock. Minutes can be one or two digits. Seconds are optional, and can be one or two digits. Milliseconds are optional, and can be up to three digits.

- A specific record in the audit trail. You specify the location of the record in terms of Audit Block Number (ABN) and digit offset within that audit block. Audit Block Numbers are absolute, starting at 1 at the beginning of the audit trail (not starting at 1 for each audit file). Audit Block Numbers and offsets can be up to 12 digits long.

For more information about rollback recovery, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

RECONSTRUCT FUNCTION

Reconstruct recovery loads selected database structures from a database backup tape and moves them forward in time to meet the rest of the database. This is the only way in which you can safely use a partial database backup tape or a portion of a full database backup tape. If you specify ONLINE, your user programs can use the database during the reconstruct recovery, as long as they do not access/update the structures that are being reconstructed. The default value is OFFLINE.

If a nonerror failure occurs during a reconstruct recovery, you can restart the failed function with the RESTART function.

For more information about reconstruct recovery, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

REORGANIZE FUNCTION

You use the REORGANIZE function to perform maintenance on or to make physical changes to database structure files. Two approaches to this process are possible.

In the first approach, you choose REORGANIZE and specify a list of structures to be generated. This initiates the generalized REORG UTILITY (DMSREO) to make the changes to the database. Garbage collection of any type of structure, set balancing/generate-new, and subset balancing can be done using this approach. (Generate-new reads the associated data set, creates a new set, and discards the existing set.) Note that structures associated with those that are specified in the generate list may also be implicitly generated as a part of the reorganization process.

If you specify ORDERED BY <set name> for a data set in the generate list, the generated data set will be ordered according to that set. Set name must be an existing set of that data set. Use this option to optimize performance when user programs access the data set primarily by the specified set.

If you do not specify ORDERED BY <set name> for a data set in the generate list, the generated data set will be in the same order as the original. In either case, the generation of a data set performs garbage collection on that structure.

If you specify FROM DATA SET for a set in the generate list, a new set will be generated from the associated data set and the original set will be discarded. This corresponds to a generate-new on that structure. You can generate only disjoint sets in this manner. You typically use this generation approach when the existing set is corrupted. Note that the order of any duplicates in the original set is lost when you use this generation approach.

If you do not specify FROM DATA SET for a set in the generate list, a replacement set will be generated from the original. This action corresponds to a garbage collection and balancing on that structure.

You cannot specify FROM DATA SET for a subset using this reorganization approach. The WHERE clause associated with the subset is not known by DMSREO, so it cannot generate the replacement subset from the data set. The second approach (REORG DBP), which is described in this section, does support this capability.

If you specify COPY TO DISK/PACK/<pack name> for the structures in the generate list, you are requesting that the replacement structures be generated on a temporary medium that is different from the medium that the original structures reside on. This is typically done when space on the original medium is limited.

If you do not specify COPY TO DISK/PACK/<pack name>, the replacement structures will be generated on the same medium as the original structures (original medium). Implicitly generated structures are usually fixed up directly (that is, a replacement structure is not generated), so this approach is normally used for them also.

If you specify COPY BACK, each replacement structure will be copied back to its original medium immediately after it is generated. Otherwise, all REORG actions will be allowed to complete first, and then all the structures will be copied back to their original medium(s).

If you specify SYNTAX, the generate parameters will be syntactically checked and DMSREO will produce a reorganization summary for your review, but no reorganize actions will actually be performed. Then, you can use the RESTART function at a later time to initiate the previously specified reorganize actions.

The second reorganization approach requires that you modify and recompile the DASDL source to describe the desired changes and bind a tailored REORG DBP (XXXREO). You then initiate the REORG DBP to make the changes to the database by choosing REORGANIZE and not specifying any structures to be generated. Record format changes, file attribute changes, garbage collection, set balancing/generate-new, and subset balancing/generate-new can be done using this approach.

For more information on this approach to reorganization, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*.

If a reorganization process does not complete successfully, you can restart it with the RESTART function. For more information, refer to the RESTART function described in this section.

It is important to recognize that the reorganization process causes structure discontinuities in the audit trail of an audited database. This causes the following recovery restrictions:

- Rebuild recovery cannot proceed through a reorganize discontinuity audit record. Therefore, you may not use a dump created before the reorganization to rebuild the database to a point after the reorganization.
- Reconstruct recovery cannot proceed through a reorganize discontinuity audit record if any of the structures being reconstructed were included in the reorganization. Therefore, you may not use a dump created before the reorganization to reconstruct any structures involved in the reorganization.
- Rollback recovery can proceed through a reorganize discontinuity audit record, but the rollback will be aborted if any update audit records corresponding to any of the structures involved in the reorganization are encountered before the stopping point of the rollback is reached.

NOTE

You should dump all of the structure files involved in the reorganization (including implicitly generated structures), the control file, the DDF, the memory file, the tailored code file (ICM), and the DBP both before and after each reorganization. If the reorganization does not complete successfully or must be restarted, you will need the dump you made before the reorganization. You will need the dump you created after the reorganization if a rebuild recovery or a reconstruct recovery that involves any of the reorganized structures is necessary.

For more information about reorganization in general, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

PRODUCTION FUNCTION

The PRODUCTION function enables you to initiate a production database program. The Production DBP must already be bound and on disk. A production DBP is a database user program bound in with a DBP to optimize its performance.

If the user program uses programmatic switches, you can insert the switch settings at this time in the form of a string of eight hex digits. The DBP places the hex string into the first eight digit positions of the program code. You should not use this function to initiate a freestanding user program that accesses a database through a Production DBP.

For more information about Production database programs, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

PURGE FUNCTION

The PURGE function enables you to remove all the user data from database structure files, leaving an empty structure that takes up the minimum amount of space on the media. It is faster than emptying the structures with conventional user program updates. The PURGE screens provide space for the names of up to 30 structures. You cannot do a purge when user programs are using the database, and you cannot purge a restart data set.

For more information about the purge function, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

LOAD FUNCTION

The LOAD function loads a full database backup tape containing previously dumped database structure files. This process leaves the database in a ready-to-run state, and recovery is not allowed. All of the structure files on the tape are loaded. If you want to use only some of the files, you must use the RECONSTRUCT function.

You can load database backup tapes created with the DUMP function of the DMSUTL utility, the IR DUMP command, or with the SYSTEM/COPY utility. The tape must be a full database backup tape created by an offline dump.

If a nonerror failure occurs during a load, you can restart the failed function with the RESTART function.

For more information about database dumping and loading, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

DUMP FUNCTION

The DUMP function dumps database files to a database backup tape. You can dump all of the files that compose the database (full dump), or you can dump specific structures (partial dump). When you select an online dump, the tape has special recovery requirements. Database user programs can be running during the dump, when you choose an online dump. Inquiry database user programs can be running during the dump, when you choose an offline dump. Update database user programs cannot be running during the dump, when you choose an offline dump.

NOTE

The DUMP function does not dump the audit files. You should dump the audit files to a separate tape with the SYSTEM/COPY utility.

For a list of the files dumped, and more information about database dumping and loading, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

RESTART FUNCTION

The RESTART function enables you to restart when a nonerror failure occurs during one of the following functions: REBUILD, RECONSTRUCT, DATABASE LOAD, or REORGANIZATION. The restart function will cause the failed function to resume from the point of failure (if possible) without having to repeat the successfully completed portions, and without requiring you to reenter the command.

You may also use the restart function to initiate a reorganization that you previously entered as a SYN-TAX run.

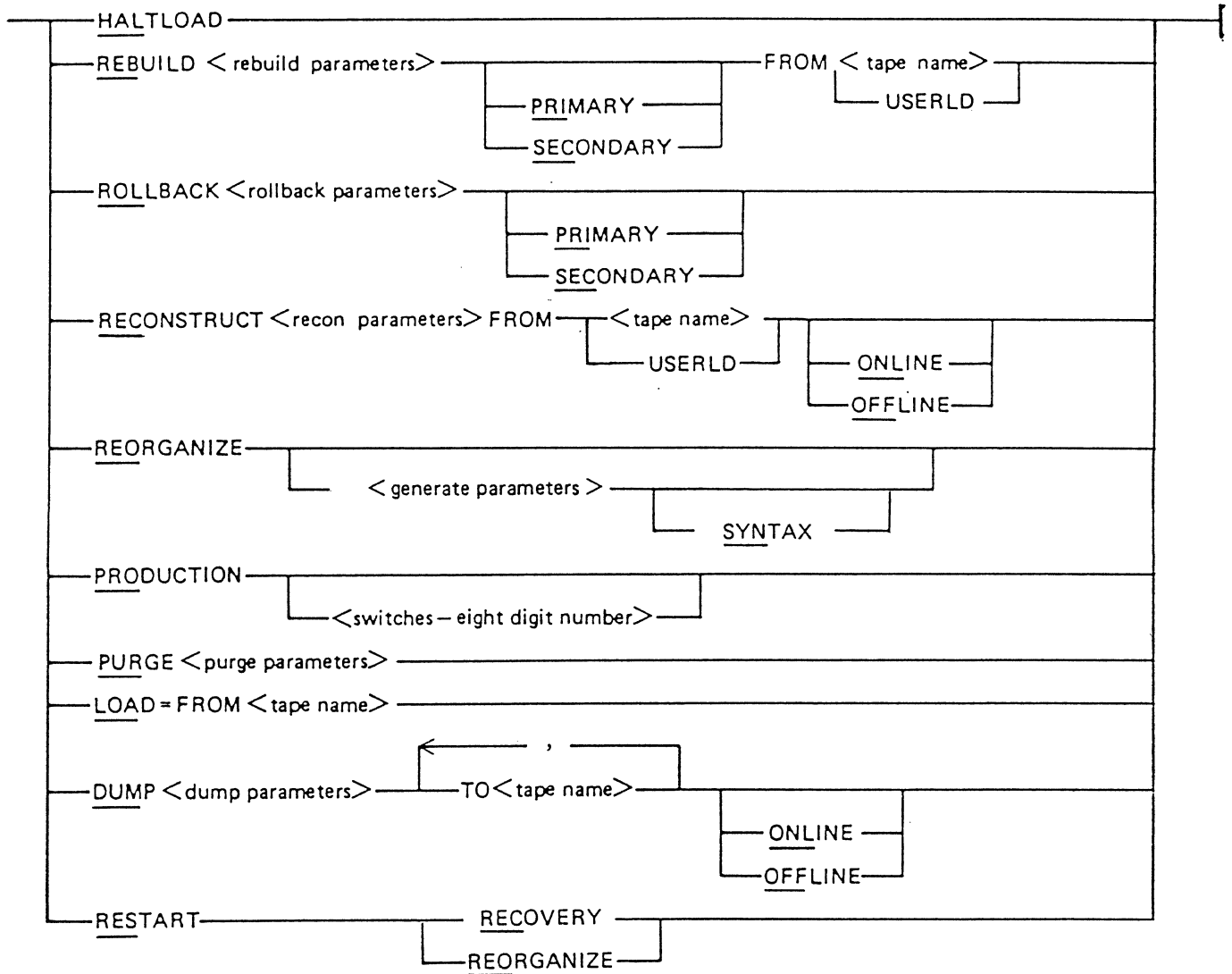
If the DBP detects (at BOJ) that a previous restartable function has failed, it will display an error indicating this and go to EOJ. If this occurs, you should use the RESTART function. The restart process will determine the last action that had completed when the failure occurred, and will display this information with a recommendation for action. If you want to restart the operation, choose the RESTART function. Otherwise, you should repeat the failed operation from the beginning.

USING DMSDLR FUNCTIONS - FREEFORM SYNTAX

The freeform syntax for using DMSDLR functions appears in Figure 10-2. It is essentially the same as the IR command syntax. You can enter this syntax on the bottom of the home menu screen. The functions correspond to the fields and choices on the DMSDLR screen. For a general description of the use of freeform syntax, refer to the "Home Menu" and "Using the DMSUTL Help Screens" portions of Section 1.

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Database Maintenance Utility (DMSDLR)

< dmsdlr freeform syntax >



< rebuild parameters >

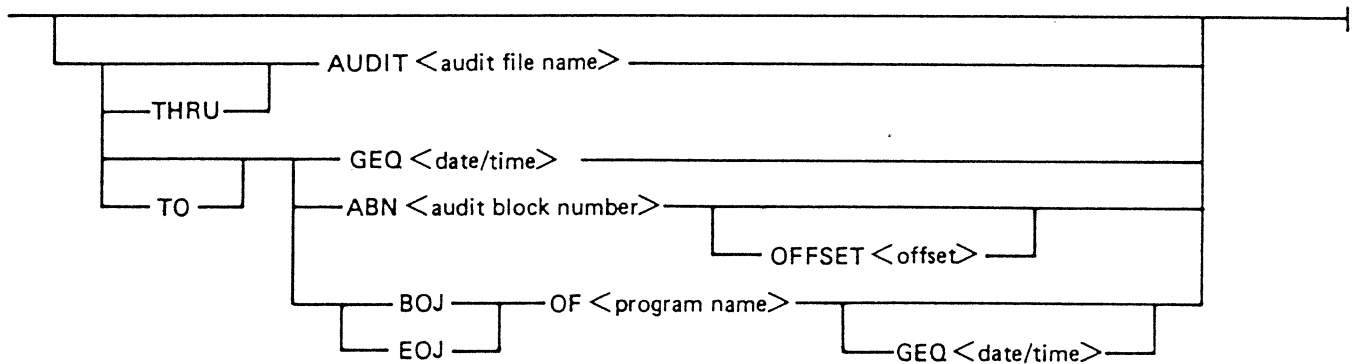
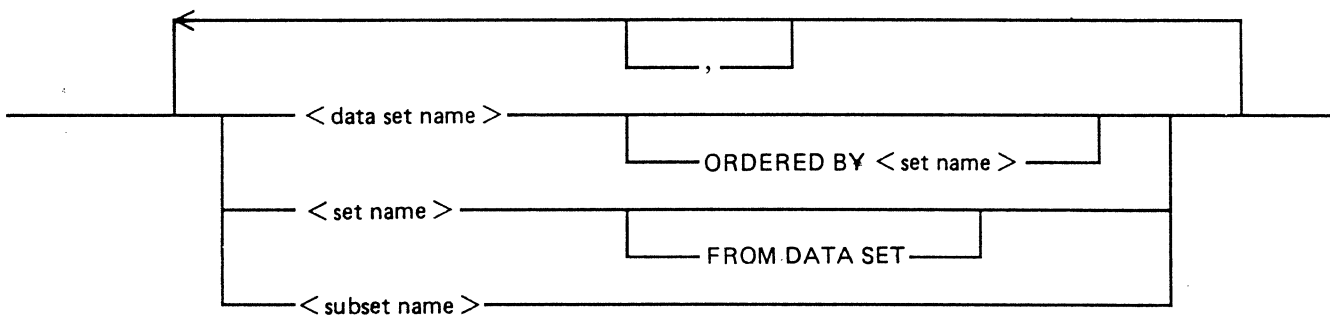


Figure 10-2. DMSUTL Freeform Syntax for DMSDLR (Sheet 1 of 4)

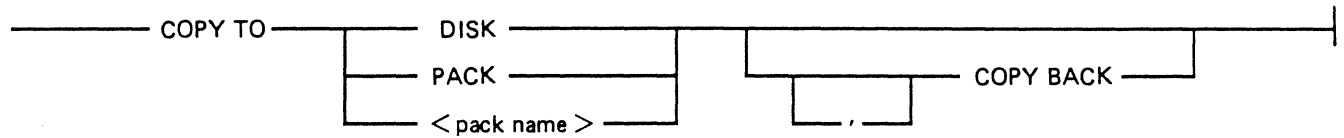


< generate parameters >

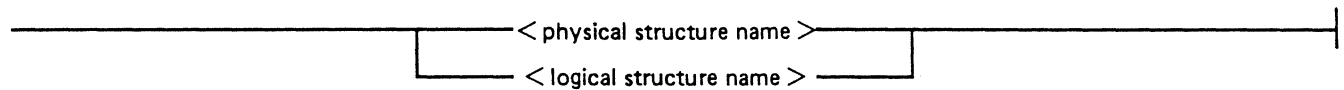


B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Database Maintenance Utility (DMSDLR)

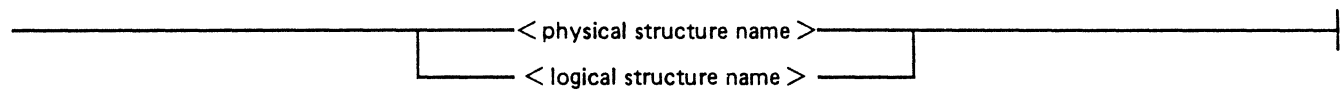
<copy parameters>



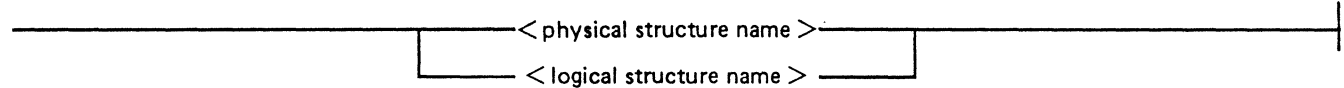
<data set name>



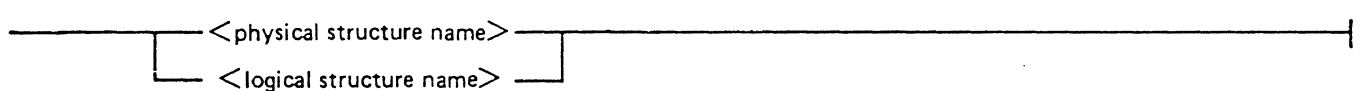
<set/subset name>



<set name>



<subset name>



<purge parameters>

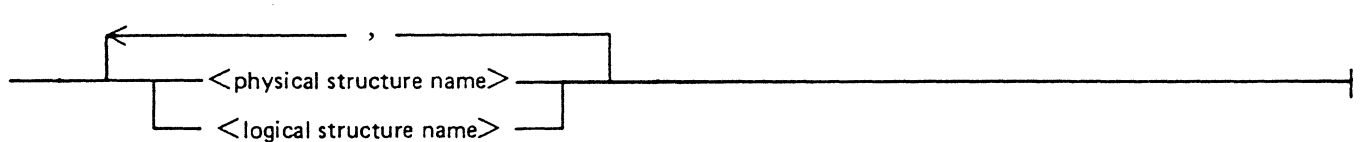
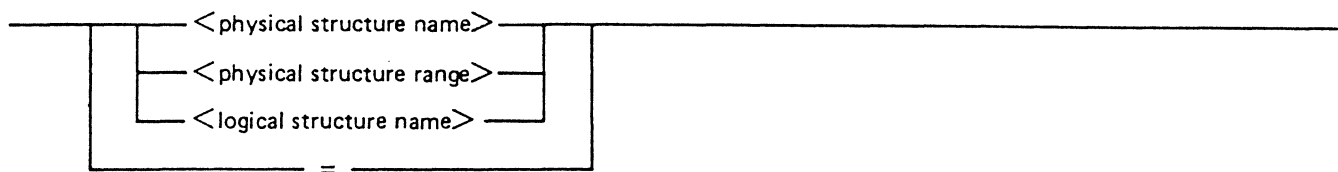


Figure 10-2. DMSUTL Freeform Syntax for DMSDLR (Sheet 3 of 4)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Database Maintenance Utility (DMSDLR)

<dump parameters>



<date/time>

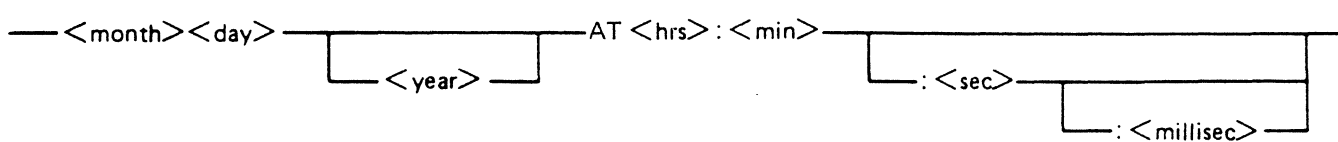


Figure 10-2. DMSUTL Freeform Syntax for DMSDLR (Sheet 4 of 4)

B 2000/B 3000/B 4000/V Series DMSII
Operations Reference Manual, Volume 3: Utilities
Database Maintenance Utility (DMSDLR)

<program name>	Program ID of program opening database. 1 to 6 characters.
<month>	JAN, FEB, etc. or JANUARY, FEBRUARY, etc.
<day>	Day of month. Up to 2 digits.
<year>	Up to 4 digits.
<hrs>	Up to 2 digits.
<min>	Up to 2 digits.
<sec>	Up to 2 digits.
<millisec>	Up to 3 digits.
<audit block number>	Up to 12 digits.
<physical structure name>	File name of physical structure file. 6 characters in form XXXNNN.
<physical structure range>	XXXMMM-XXXNNN, where XXXMMM and XXXNNN are structure names and MMM < NNN. The names are separated by a hyphen.
<logical structure name>	Structure name as specified in DASDL source. 1 to 17 characters.

INITIATING DMSDLR - STAND-ALONE MODE

You can initiate DMSDLR with the IR command. For more information about the IR command, see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

SECTION 11

UTILITY MESSAGES

The Data Management System II (DMSII) utilities produce the messages that appear in this section. DMSII produces three types of messages: nonfatal and fatal error messages, and information only system messages. Nonfatal error messages indicate that there is an error condition, but the DBP will continue to run the user program. Fatal error messages indicate that there is an error and the DBP will stop the user program immediately. Information only system messages let you know what the DBP is doing.

ABN AND OFFSET MUST CONTAIN ONLY DIGITS

This is a nonfatal error message from DMSUTL.

ABN AND RRN MUST BE ALL DIGITS

This is a nonfatal error message from DMSUTL about DMSAUD.

AN INCORRECT SCREEN HAS BEEN USED

This is a nonfatal error message from DMSUTL.

AUDIT FILE NAME : <input word> DOESN'T MATCH

AUDIT TRAIL SELECTION : <input word>

This is a nonfatal error message from DMSUTL.

AUDIT FILE NAME MUST BEGIN WITH <database name>

This is a nonfatal error message from DMSUTL.

AUDIT FILES <audit file A> <input word> AND

<audit file B> <input word> DON'T MATCH

This is a nonfatal error message from DMSUTL about DMSAUD.

AUTO TAPE LOAD REQUIRED SINCE DDF, CNF FILES NOT PRESENT

This is a nonfatal error message from DMSUTL about DMSDLR.

BEGINNING ABN EXCEEDS ENDING ABN

This is a nonfatal error message from DMSUTL about DMSAUD.

BEGINNING AUDIT FILE EXCEEDS ENDING AUDIT FILE

This is a nonfatal error message from DMSUTL about DMSAUD.

BLOCK RANGE TRUNCATED TO: <number>

This is a nonfatal error message from DMSUTL.

BLOCK RANGE VALUE EXCEEDS 8-DIGIT LIMIT

This is a nonfatal error message from DMSUTL.

CAN ONLY DO "IR REORGANIZE" ON THE REORGANIZE DBP

You have made an attempt to use the Reorg DBP for a recovery. The Reorg DBP does not perform recovery functions. If the reorganization process fails, you must repeat it with the "IR <XXXREO> REORGANIZE" command or the DMSUTL utility after you have solved the problem. This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND AUDIT BLOCK NUMBER <input word>

This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND AUDIT FILE NAME <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND AUDIT TRAIL RANGE
This is a nonfatal error message from DMSUTL about DMSAUD.

CANNOT UNDERSTAND BLOCK: <input word>
This is a nonfatal error message from DMSUTL about DMSAUD.

CANNOT UNDERSTAND DAY: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND DUMP PARAMETER: <input word>
This is a nonfatal error message from DMSUTL about DMSDLR.

CANNOT UNDERSTAND IV NUMBER: <input word>
This is a nonfatal error message from DMSUTL about DMSDLR.

CANNOT UNDERSTAND LOAD PARAMETER: <input word>
This is a nonfatal error message from DMSUTL about DMSDLR.

CANNOT UNDERSTAND MILLISECS: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND MINUTES
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND MONTH: [<input word>]
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND OPTION: <input word>
This is a nonfatal error message from DMSUTL about DMSDLR.

CANNOT UNDERSTAND RELATIVE RECORD NUMBER: <input word>
This is a nonfatal error message from DMSUTL about DMSAUD.

CANNOT UNDERSTAND SECONDS: <input word>
This is a nonfatal error message from DMSUTL about DMSDLR.

CANNOT UNDERSTAND STRUC: <input word>
This is a nonfatal error message from DMSUTL about DMSAUD.

CANNOT UNDERSTAND SYNTAX. EXPECTED: <syntax>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE COMMAND OPTION
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DATA SET COMMAND
This is a nonfatal error message from DMSUTL about DMSGEN.

CANNOT UNDERSTAND THE DBPDMP DUMP FILE NUMBER: <input word>
This is a nonfatal error message from DMSUTL about DBPDMP.

CANNOT UNDERSTAND THE DBPDMP OPTION: <input word>
This is a nonfatal error message from DMSUTL about DBPDMP.

CANNOT UNDERSTAND THE DBPDMP PROCESSOR NUMBER: <input word>
This is a nonfatal error message from DMSUTL about DBPDMP.

CANNOT UNDERSTAND THE DDFLIS COMMAND: <input word>
This is a nonfatal error message from DMSUTL about DDFLIS.

CANNOT UNDERSTAND THE DMSAUD COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DMSCTL COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DMSDAP COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DMSDLR COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DBPDMP COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DMSGEN COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DMSLST COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND THE DMSTAT COMMAND: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND TYPE: <input word>
This is a nonfatal error message from DMSUTL.

CANNOT UNDERSTAND YEAR: <input word>
This is a nonfatal error message from DMSUTL about DMSDLR.

CHECKSUM ERROR IN FILE: <filename> AT BLOCK: <blocknumber>
This is a nonfatal error message from DMSUTL. You should check the integrity of the results.

CONTROL AND SUMMARY REPORTS NOT AVAILABLE WITH ISAM FILES
This is an information only message from DMSUTL.

CONTROL FILE IS NEEDED FOR PURGE BUT IS NOT PRESENT
This is a nonfatal error message from DMSUTL about DMSDLR.

CONTROL FILE IS NEEDED FOR RECONSTRUCT BUT IS NOT PRESENT
This is a nonfatal error message from DMSUTL about DMSDLR.

DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.
Use IR RESTART command or RESTART option of DMSUTL utility to recover.

DATABASE NAME EXCEEDS 6-CHARACTER LIMIT: <input word>

This is a nonfatal error message from DMSUTL.

DATABASE NAME HAS FEWER THAN 3 CHARACTERS: <input word>

This is a nonfatal error message from DMSUTL.

DATABASE NAME PARAMETERS ARE INCOMPLETE

This is a nonfatal error message from DMSUTL.

DAY, YEAR, AND ALL TIME FIELDS ALLOW DIGITS ONLY

This is a nonfatal error message from DMSUTL about DMSDLR.

DBPDMP IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DBPDMP CANNOT BE EXECUTED USING DMSUTL FROM THE SPO

You need to either execute DMSUTL from a timesharing terminal or execute DBPDMP directly.

DDFLIS IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DMSAUD IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DMSAUD CANNOT BE EXECUTED USING DMSUTL FROM THE SPO

This is a nonfatal error message from DMSUTL. Either execute DMSUTL from a timesharing terminal or execute DMSAUD directly.

DMSCTL IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DMSDAP IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DMSDAP PARAMETERS ARE MISSING

This is a nonfatal error message from DMSUTL about DMSDAP.

DMSDLR IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DMSDLR CANNOT BE EXECUTED USING DMSUTL FROM THE SPO

This is a nonfatal error message from DMSUTL. Execute DMSUTL from a timesharing terminal or execute the desired function directly by using an IR command.

DMSTAT IS NOW EXECUTING IN BATCH MODE

This is an information only message from DMSUTL.

DMSUTL IS UNABLE TO CONTINUE — BYE

This is a fatal error message from DMSUTL that indicates DMSUTL has encountered a corrupt or missing file.

DUMP FILE NUMBER MUST CONTAIN ONLY DIGITS

This is a nonfatal error message from DMSUTL about DBPDMP.

DUMP REQUEST DENIED, RECOVERY IN PROGRESS

This is a nonfatal error about the DUMP function of DMSDLR utility that you initiated with the IR DUMP command or DUMP option of DMSUTL utility. The database cannot be dumped while a recovery is in progress. Wait until the recovery has finished; then repeat the DUMP request.

ENTER DMSDAP PARAMETERS

This is a prompt for the AX command from DMSUTL.

ENTER DMSGEN PARAMETERS

This is a prompt for the AX command from DMSUTL.

ENTER DMSLST PARAMETERS

This is a prompt for the AX command from DMSUTL.

ENTER DMSTAT PARAMETERS

This is a prompt for the AX command from DMSUTL.

ENTER DMSUTL PARAMETERS

This is a prompt for the AX command from DMSUTL.

ENTER DMSUTL PARAMETERS OR "HELP"

This is a prompt for the AX command from DMSUTL.

EXECUTION OF THE ZIPPED DMSUTL HAS STOPPED DUE TO INTERNAL ERROR

This is a fatal error message from DMSUTL. DMSUTL caused the initiation of another DMS utility program. That utility program encountered a corrupt file and discontinued.

FAMILYNAME EXCEEDS 6-CHARACTER LIMIT: <input word>

This is a nonfatal error message from DMSUTL.

FOR DAY/TIME: ENTER AT LEAST MONTH, DAY, HOURS, AND MINUTES

This is a nonfatal error message from DMSUTL about DMSDLR.

FORMAT LEVEL OF CONTROL FILE IS INCOMPATIBLE WITH THIS RELEASE

This is a nonfatal error message from DMSUTL.

FORMAT LEVEL OF DDF IS INCOMPATIBLE WITH THIS RELEASE

This is a nonfatal error message from DMSUTL.

FORMAT LEVEL OF ISAM FILE IS INCOMPATIBLE WITH THIS RELEASE

This is a nonfatal error message from DMSUTL about DMSLST.

FUNCTION NOT AVAILABLE: <input word>

This is an information only message from DMSUTL.

HOST LANGUAGE: <source file> IS BEING CREATED IN BATCH MODE

This is an information only message from DMSUTL. It indicates that the DMSUTL program has initiated the DMSGEN function.

INCOMPATIBLE CONTROL FILE MEDIA TYPE

This is an information only message from DMSUTL.

INCOMPATIBLE STATISTICS FILE MEDIA TYPE

This is a nonfatal error message from DMSUTL.

INVALID BLOCK RANGE — EOF BLOCK IS: <input word>

This is an information only message from DMSUTL.

INVALID OR MISSING BLOCK RANGE VALUE

This is an information only message from DMSUTL.

ISAM FILE NAME MUST BE 6 CHARACTERS LONG: <input word>

This is a nonfatal error message from DMSUTL about DMSLST.

KEY NUMBER IS TOO LARGE: <input word>

This is a nonfatal error message from DMSUTL about DMSLST.

LOGICAL/PHYSICAL NAMES CLASH. DMSDLR EXECUTED USING PHYSICAL NAME.

This is a nonfatal error message from DMSUTL about DMSGEN.

MAXIMUM LENGTH OF <item> IS: <number> CHARACTERS

This is a nonfatal error message from DMSUTL.

MAXIMUM LENGTH OF <item> IS: <number> DIGITS

This is a nonfatal error message from DMSUTL.

MAXIMUM OF NINE "OR" CASES ALLOWED

This is a nonfatal error message from DMSUTL about DMSAUD.

MISSING A DASH IN BLOCK RANGE

This is a nonfatal error message from DMSUTL about DMSLST.

MISSING AUDIT TRAIL RANGE

This is a nonfatal error message from DMSUTL about DMSAUD.

MISSING "END" AFTER LAST STRUCTURE

This is a nonfatal error message from DMSUTL about DMSGEN.

MISSING FAMILYNAME

This is an information only message from DMSUTL.

MISSING RESTART DATA SET NAME

This is a nonfatal error message from DMSUTL about DMSGEN.

NO PHYSICAL FILE FOR ACCESSES

This is a nonfatal error message from DMSUTL about DMSLST.

NO PHYSICAL FILE FOR REMAPS

This is a nonfatal error message from DMSUTL about DMSLST.

NO PREV LOAD/RECOVERY INFORMATION IS AVAILABLE

This is a nonfatal error message from DMSUTL about the RESTART option of DMSDLR.

NOTHING WAS RECEIVED. PLEASE REENTER COMMAND

This is a nonfatal message from DMSUTL.

OPTION HAS ALREADY BEEN CHOSEN: <input word>

This is a nonfatal error message from DMSUTL about DMSAUD.

PHYSICAL FILE NOT FOUND: <file> on <media>

This is a nonfatal error message from DMSUTL.

PLEASE CHOOSE AT LEAST ONE REPORT

This is a nonfatal error message from DMSUTL about DMSLST.

PLEASE CHOOSE AT LEAST ONE STRUCTURE

This is a nonfatal error message from DMSUTL about DMSDLR.

PLEASE CHOOSE EITHER ABN OR RRN

This is a nonfatal error message from DMSUTL about DMSAUD.

PLEASE CHOOSE EITHER BOJ POINT OR EOJ POINT

This is a nonfatal error message from DMSUTL about DMSDLR.

PLEASE CHOOSE EITHER ONLINE OR OFFLINE DUMP

This is a nonfatal error message from DMSUTL about DMSDLR.

PLEASE CHOOSE EITHER PRIMARY OR SECONDARY AUDIT TRAIL

This is a nonfatal error message from DMSUTL about DMSAUD.

PLEASE ENTER A TAPE NAME

This is a nonfatal error message from DMSUTL about DMSDLR.

PLEASE FILL IN AT LEAST ONE DATA SET NAME

This is a nonfatal error message from DMSUTL about DMSGEN.

PLEASE FILL IN AT LEAST THE FIRST DATA SET NAME

This is a nonfatal error message from DMSUTL about DMSGEN.

PLEASE FILL IN DUMP FILE NUMBER

This is a nonfatal error message from DMSUTL about DBPDMP.

PLEASE FILL IN FIRST THREE CHARS OF DATABASE NAME

This is a nonfatal error message from DMSUTL.

PLEASE FILL IN PROGRAM-ID AREA

This is a nonfatal error message from DMSUTL about DMSDLR.

PLEASE FILL IN THE DATABASE NAME AREA

This is a nonfatal error message from DMSUTL.

PLEASE MAKE ALL DESIRED CHANGES

This is a nonfatal error message from DMSUTL about DMSGEN.

PLEASE SELECT A COMMAND/OPTION FROM 1 TO n

This is a nonfatal error message from DMSUTL.

PLEASE SELECT A FUNCTION FROM 0 TO 9

This is an information only message from DMSUTL.

PREV LOAD/RECOVERY MAY BE INCOMPLETE — USE RESTART TO INVESTIGATE

This is a nonfatal error message from DMSUTL about DMSDLR.

PRIMARY ID FOR DDF IS UNRECOGNIZABLE

This is an information only message from DMSUTL.

PRINT FILE IS NOT OPEN

This is an information only message from DMSUTL.

PRINT REMINDER: PRINTER OUTPUT WILL STOP WHENEVER TERMINAL OUTPUT IS STOPPED.

This is an information only message from DMSUTL about DMSTAT or DMSLST. When you select the PRINT option, only that part of the report that you viewed from the terminal is put into the printer backup file. To get the whole report without viewing all of it from the terminal, use the PRINTONLY option.

PRINT REMINDER: PRINTER OUTPUT WILL FINISH EVEN IF TERMINAL OUTPUT IS STOPPED

This is an information only message from DMSUTL about DMSCTL or DDFLIS. Whether you select the PRINT or PRINTONLY option, the utility program puts the entire report into the printer output file.

PROGRAM ID EXCEEDS 6-CHAR LIMIT: <input word>

This is a nonfatal error message from DMSUTL about DMSDLR.

REORG STRUCTURE LOCKED IN THE CONTROL FILE

A structure that you want to reorganize is corrupted and locked. Recover the structure before you reorganize.

REORGANIZATION NOT RESTARTABLE; RELOAD DATABASE FROM THE PREVIOUS DUMP

The failed reorganization process is not restartable because of some inconsistency. Reload your database files from the dump you made before the reorganization.

RESTART DATA SET NOT A DATA SET; HOST LANGUAGE SYNTAX ERROR MAY RESULT

This is a nonfatal error message from DMSUTL about DMSGEN.

RESTART DATA SET NOT DISJOINT; HOST LANGUAGE SYNTAX ERROR MAY RESULT

This is a nonfatal error message from DMSUTL about DMSGEN.

RESTART DATA SET NOT FOUND; HOST LANGUAGE SYNTAX ERROR MAY RESULT

This is a nonfatal error message from DMSUTL about DMSGEN.

RESTART DATA SET NOT INCLUDED; HOST LANGUAGE SYNTAX ERROR MAY RESULT

This is a nonfatal error message from DMSUTL about DMSGEN.

RESTART DATA SET TOO SMALL; HOST LANGUAGE SYNTAX ERROR MAY RESULT

This is a nonfatal error message from DMSUTL about DMSGEN.

SECONDARY ID FOR DDF IS UNRECOGNIZABLE

This is an information only message from DMSUTL.

SELECT OPTION CANNOT BE USED WITH PRINT, VERIFY, OR COPY

This is a nonfatal error message from DMSUTL about DMSAUD.

STATISTICS NOT SET ON DATABASE

This is an information only message from DMSUTL.

STATISTICS STRUCTURE IS EMPTY

This is an information only message from DMSUTL.

STATS FILE TIMESTAMP DOESN'T MATCH CONTROL FILE TIMESTAMP

This is a nonfatal error message from DMSUTL about DMSTAT.

STRUCTURE: <structure> HAS ALREADY BEEN SPECIFIED

This is a nonfatal error message from DMSUTL.

STRUCTURE: <structure> IS EMPTY

This is a nonfatal error message from DMSUTL.

STRUCTURE: <structure> IS LOCKED OUT

This is a nonfatal error message from DMSUTL.

STRUCTURE: <structure> IS NOT A DATA SET

This is a nonfatal error message from DMSUTL.

STRUCTURE: <structure> IS NOT DISJOINT

This is a nonfatal error message from DMSUTL.

STRUCTURE: <structure> IS OPEN

This is a nonfatal error message from DMSUTL.

STRUCTURE: <structure> IS UNKNOWN TO THE CONTROL FILE

This is a nonfatal error message from DMSUTL.

STRUCTURE LIST LIMIT FROM SPO IS 3

This is an information only message from DMSUTL.

STRUCTURE NAME EXCEEDS 17-CHAR LIMIT: <structure name>

This is a nonfatal error message from DMSUTL about DMSDAP.

STRUCTURE NAME IS MISSING OR UNRECOGNIZABLE

This is a nonfatal error message from DMSUTL.

STRUCTURE NUMBER: <structure number> HAS BEEN DELETED

This is a nonfatal error message from DMSUTL.

STRUCTURE NUMBER: <structure number> IS UNASSIGNED

This is a nonfatal error message from DMSUTL.

STRUCTURE NUMBER CAN ONLY BE 3 DIGITS LONG

This is a nonfatal error message from DMSUTL.

STRUCTURE NUMBER EXCEEDS HIGHEST STRUCTURE

This is an information only message from DMSUTL.

STRUCTURE NUMBER MUST BE ALL DIGITS

This is an information only message from DMSUTL.

STRUCTURE REPORT HAS ALREADY BEEN SPECIFIED

See the message "THE STRUCTURE REPORT HAS ALREADY BEEN SPECIFIED."

SWITCHES MUST BE IN THE RANGE: 0-9 OR A-F

This is a nonfatal error message from DMSUTL about DMSDLR.

TAPE NAMES MUST BE ALPHANUMERIC AND BEGIN WITH A TO Z

This is a nonfatal error message from DMSUTL about DMSDLR that indicates the DMSUTL program has initiated the DMSGEN function.

THE HOST LANGUAGE SOURCE: <source file> IS BEING CREATED IN BATCH MODE

This is an information only message from DMSUTL.

THE ONLY UTILITY WITH ISAM CAPABILITY IS DMSLST

This is an information only message from DMSUTL.

THE REINITIALIZE OPTION GENERATES ALL REPORTS

This is an information only message from DMSUTL.

THE REINITIALIZE OPTION NEEDS TO USE ALL STRUCTURES

This is an information only message from DMSUTL.

THE REQUESTED SCREEN WAS NOT FOUND IN THE SCREEN FILE

This is a FATAL error message from DMSUTL. The screen file associated with DMSUTL has become corrupted.

THE STRUCTURE REPORT HAS ALREADY BEEN GENERATED

This is a nonfatal error message from DMSUTL about DMSTAT.

TIMESTAMP MISMATCH: ENTER "GO" TO OVERRIDE, OTHERWISE "STOP"

This message comes from DMSTAT when you run a program in ODT or card input mode. The timestamps of the control file and the statistics file don't match. To print an old statistics file, enter "AX <mix number> GO".

TIMESTAMP MISMATCH: STATS FILE, CONTROL FILES. RETRANSMIT TO OVERRIDE

This message comes from DMSTAT when you run a program with DMSUTL. The timestamps of the control file and the statistics file don't match. To print an old statistics file, transmit from "home."

UNRECOGNIZABLE MEDIA TYPE FOR: <file>

This is a nonfatal error message from DMSUTL.

INDEX

A

audit trail records, 10-3

B

B 774 Message Control System Generator (MC7), 1-5

C

CANDE, 1-1

Common Screen Choices

- home option, 1-11
- more-help option, 1-10
- next-form option, 1-11
- previous-form option, 1-11
- quit option, 1-11
- refresh option, 1-10-11

control file, 3-3, 5-1

COPY BACK, 10-5

COPY TO, 10-5

D

Data Definition File (DDF), 1-1, 1-16, 7-1

Database

- STATISTICS option
- SET, 3-1

Database Specifications

- using DMSUTL in batch mode, 1-16

DBPDMP, 1-1, 2-1

- examples, 2-6
- executing from cards or card images, 2-5
- executing from the ODT, 2-5
- freeform syntax, 2-4
- initiating from a WFL job, 2-5
- initiating in screen mode, 2-1
- initiating in stand-alone mode, 2-5
- initiating with PM command, 2-5
- menu screen, 2-2
- options, 2-3

DDFLIS, 1-1, 7-1

- freeform syntax, 7-2
- initiating in batch mode, 7-3
- initiating in screen mode, 7-1
- menu screen, 7-1
- specification for batch use, 7-3

DMSAUD, 1-1, 8-1

- audit record types, 8-5
- description of consistency analysis, 8-17
- executing from cards or card images, 8-7
- executing from the ODT, 8-7
- freeform syntax, 8-6
- initiating from a WFL job, 8-7
- initiating in screen mode, 8-1
- initiating in stand-alone mode, 8-7
- menu screen, 8-2
- parameters for batch use, 8-8
- PRINT option, 8-10
- record selection, 8-4
- record selection screen, 8-4
- reports, 8-11

VERIFY option, 8-10, 8-17

DMSCTL, 1-1, 5-1

- freeform syntax, 5-2
- initiating in batch mode, 5-3
- initiating in screen mode, 5-1
- menu screen, 5-1
- report, 5-3
- specification for batch use, 5-3

DMSDAP, 1-1, 6-1

- description of analysis, 6-4
- EXTENDED VERIFY, 6-3
- freeform syntax, 6-3
- initiating in batch mode, 6-3
- initiating in screen mode, 6-1
- menu screen, 6-2
- Report, 6-5
- specification for batch use, 6-3
- VERIFY, 6-3

DMSDLR, 1-1, 10-1

- DUMP function, 10-8
- initiating in stand-alone mode, 10-13
- LOAD function, 10-7
- menu screen, 10-2
- PRODUCTION function, 10-7
- PURGE function, 10-7
- REORGANIZE function, 10-5
- RESTART function, 10-4, 10-8

DMSDLR functions

- initiating in screen mode, 10-2
- using
 - with freeform syntax, 10-8

DMSGEN, 1-1, 9-1

- formatting flat files with UNLOAD, 9-6
- freeform syntax, 9-3
- initiating in batch mode, 9-4
- initiating in screen mode, 9-1
- initiating LOAD, 9-6
- initiating SKELETON, 9-7
- initiating UNLOAD, 9-5
- LOAD function, 9-6
- menu screen, 9-2
- parameters for batch use, 9-4
- SKELETON function, 9-7
- specification for batch use, 9-4
- UNLOAD function, 9-5

DMSLST, 1-1, 4-1

- BCI Report, 4-13
- Block One Report, 4-14
- Block Report, 4-11
- Control Report, 4-10
- freeform syntax, 4-4
- ISAM freeform syntax, 4-8
- ISAM Menu Screen, 4-6
- ISAM options, 4-7
- listing database files in batch mode, 4-4
- listing database files in screen mode, 4-1
- listing ISAM files in batch mode, 4-8
- listing ISAM files in screen mode, 4-6
- menu screen, 4-2

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual living or otherwise, or that of any group or association is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the CLASS specified as 2 (S.W.:System Software), and the Type specified as 3 (DOC), and the product specified as the 7-digit form number of the manual (for example, 5026586).